COMPUTERGESTÜTZTE ANALYSE DIGITALER VIDEOSEQUENZEN

Schriftliche Hausarbeit im Rahmen der ersten Staatsprüfung für das Lehramt für die Sekundarstufe I/II

dem Staatlichen Prüfungsamt für erste Staatsprüfungen für Lehrämter an Schulen, Essen

vorgelegt von	**:**	**Thomas Kersting**
		Kückelmannshof 23
		45894 Gelsenkirchen
Tag der Abgabe	**:**	**07.10.1997**
Hochschulort	**:**	**Essen**
Themensteller	**:**	**Prof. Dr. H.-J. Schlichting**
		Fachbereich 7
		Didaktik der Physik

INHALTSVERZEICHNIS

EINLEITUNG 4

1 VORSTELLUNG BEREITS EXISTIERENDER SYSTEME 6
1.1 Orvico 6
1.2 Carmen 6
1.3 Analyse von Digitalvideos 7

2 EINFÜHRUNG IN DIE GRUNDLAGEN DER VIDEODIGITALISIERUNG 8
2.1 Technische Grundbegriffe 9
2.1.1 Die Videotechnik und ihre Normen 9
2.1.1.1 Auflösung 10
2.1.1.2 Bildwiederholfrequenzen 10
2.1.1.3 Videosignale 11
2.1.1.4 FBAS, YC, YUV 12
2.1.2 Kompression 12
2.1.2.1 YUV 4:2:2 15
2.1.2.2 Motion - JPEG 15
2.1.2.3 MPEG 16
2.1.2.4 Fraktale Kompression 18
2.1.2.5 Software - Codecs 19
2.2 Hardware 20
2.2.1 Systemanforderungen 20
2.2.1.1 Monitor und Grafikkarte 21
2.2.1.2 Festplatte 21
2.2.2 Digitalisierungskarten 24
2.3 Digitalisierungssoftware 25

3 DIE AUFNAHME VON VIDEOSEQUENZEN 27
3.1 Versuchsaufbau 27
3.2 Farbmarkierungen 29
3.2.1 Größe und Form 30
3.2.2 Farbe 31
3.2.3 Material 31
3.3 Kameraeinstellungen 31
3.3.1 Zoom 32
3.3.2 Verschlußzeiten 33
3.3.3 Belichtung 34
3.3.4 Autofokus 34
3.3.5 Weißabgleich 35
3.3.6 Zusammenfassung 35
3.4 Softwareeinstellungen 36
3.4.1 Bildrate 36
3.4.2 Farbsättigung 37
3.4.3 M-JPEG- Kompressionsfaktor 40
3.4.4 Systemeinstellungen 41

4 BESCHREIBUNG DES PROGRAMMS "VIANA" 43

4.1 Entwicklung 43

4.1.1 Die Videodarstellung 44

4.1.2 Auslesen von Farbinformationen 45

4.1.3 Die Suche nach bestimmten Farben 46

4.1.4 Zweitsuche 49

4.1.5 Mittelpunktsberechnung 49

4.1.6 Farbübernahme 50

4.1.7 Suche beschleunigen 51

4.1.8 Berechnung von Geschwindigkeiten und Beschleunigungen 52

4.1.9 Diagrammanzeige 53

4.1.10 Datenexport 54

4.1.11 Stroboskopbild 55

4.1.12 Manuelle Positionserfassung 56

4.2 Bedienung 56

4.2.1 Film laden, Bedienung des Mediaplayers 57

4.2.2 Automatische Positionserfassung 59

4.2.2.1 Farbanzeige 59

4.2.2.2 Anzeige des Suchergebnisses 60

4.2.2.3 Sucheinstellungen 61

4.2.2.4 Suchgebiet einschränken 65

4.2.2.5 Kalibrieren 65

4.2.3 Manuelle Positionserfassung 66

4.2.4 Auswertung 66

4.2.4.1 Diagramme 67

4.2.4.2 Diagrammformat 68

4.2.4.3 Datenexport 70

4.2.5 Stroboskopbild 71

5 VERSUCHSDURCHFÜHRUNGEN 73

5.1 Der schräge Wurf 75

5.2 Das Federpendel 82

5.3 Die schiefe Ebene 85

5.4 Die Verfolgung mehrerer Objekte 88

6 SCHLUßWORT 90

LITERATURANGABEN 91

ANHANG

A : Quelltext 93

B : Schritt - für - Schritt - Anleitung 126

C : Internetadressen 127

Einleitung

„Die Experimentalphysik gewinnt durch exakte Beobachtung des Naturgeschehens (...) Kenntnis über die qualitativen und quantitativen Zusammenhänge der Meßgrößen.“ [1]

Die ursprünglichste Art dieser Beobachtung ist sicherlich das einfache Hinschauen. Dadurch können - besonders bei Bewegungsvorgängen - oftmals Vermutungen über zugrundeliegende Naturgesetze getroffen werden. Um diese Vermutungen zu bestätigen und letztendlich zu physikalischen Gesetzen zu gelangen, eignet sich diese natürliche Meßmethode jedoch kaum.

Durch ein ‘technisches Auge’, also einen Fotoapparat oder eine Videokamera, läßt sich die Position eines bewegten Körpers und der dazugehörige Zeitpunkt sehr genau bestimmen. So kann eine Bewegung gefilmt und auf einem Fernseher in Zeitlupe dargestellt werden. Auf einer Klarsichtfolie, die auf den Bildschirm gelegt wird, könnte so die Position verfolgt werden. Eine mitgefilmte Demonstrationsuhr liefert die jeweilige Zeit.

Da dieses Verfahren für die Schule sehr zeitraubend und ungenau ist, erscheint eine andere Auswertung des erzeugten Videomaterials sinnvoll zu sein.

Neue technische Entwicklungen im Computerbereich lassen die Digitalisierung von Videofilmen in sehr guter Qualität zu. In dieser digitalen Form ist eine komfortable Darstellung und Untersuchung der Bildinformationen möglich.

Dazu existieren Programme wie *Diva*, *Galileo* u.a., die speziell zur Positionserfassung eines Objektes in digitalen Videos entwickelt wurden. Bei diesen Programmen muß der Anwender die Position jedoch in jedem Bild selbst angeben. Wenige Sekunden eines Videofilmes können bereits aus über 100 Einzelbildern bestehen, eine Auswertung von Filmen dieser Länge erweist sich als sehr aufwendig.

Da sich nach der Digitalisierung eines Filmes alle Informationen über die Objektposition schon im Rechner befinden, liegt es nahe, den Computer selbständig nach dieser Position suchen zu lassen. Dazu ist ein Programm erforder-

---

[1] Meyers großes Taschenlexikon

lich, das ein Objekt in den Videobildern automatisch erkennt. Die vorliegende Examensarbeit stellt sich der Aufgabe, ein solches Computerprogramm zu entwickeln.

Im ersten Kapitel soll ein kurzer Überblick über bereits existierende Programme auf dem Gebiet der Videodigitalisierung und - auswertung gegeben werden.
Im zweiten Kapitel werden die technischen Grundlagen der Videodigitalisierung behandelt.
Eine automatische Positionserfassung stellt gewisse Anforderungen an das zu untersuchende Video. Kapitel 3 hat die Erzeugung von Videosequenzen, die für eine automatische Analyse geeignet sind, zum Thema.
Kapitel 4 beschäftigt sich mit dem im Rahmen dieser Arbeit entwickelten Programm '**Viana**' und seiner Bedienung, bevor es in Kapitel 5 an einigen Versuchen vorgeführt wird.

1 Vorstellung bereits existierender Systeme

Zur Erfassung einer Objektposition mit Hilfe von Video und Computer kann man prinzipiell zwei Systeme unterscheiden. Bei dem einen wird durch eine spezielle Hardware das Videosignal untersucht und nur die Objektposition an den Computer übermittelt. Bekannt geworden sind die Systeme Orvico und Carmen.
Die zweite Methode speichert sämtliche Informationen des Videosignals im Computer und untersucht diese in einem zweiten Arbeitsschritt (Kap.1.3.).

1.1 Orvico

Mit Orvico[2] ist eine *Onlineverfolgung* des hellsten Punktes eines Videobildes möglich. Dazu muß eine spezielle Computerkarte in den PC integriert werden.
Das Videokamerasignal enthält Farb- und Helligkeitsinformationen von jeder Zeile eines Videobildes. Die Helligkeitsinformationen werden von der Computerkarte untersucht. Sollte das Signal eine einstellbare Helligkeitsschwelle überschreiten, so kann die Schaltung über die Zeilenzahl des Bildes die vertikale Koordinate festlegen. Die horizontale Koordinate errechnet sich über die Zeit, die seit dem Beginn des Zeilenaufbaus bis zum Erreichen der Schwelle vergangen ist. Von der Karte werden anschließend nur die Koordinaten der Schwellenüberschreitung an den Computer übermittelt.
Als Markierung des Objektes dient je nach Beleuchtung ein weißer Farbfleck oder eine kleine Lampe.[1]

1.2 Carmen

Mit Carmen[3] läßt sich der Mittelpunkt eines hellen Objektes bestimmen und online verfolgen. Es arbeitet mit einer speziellen Digitalisierungskarte. Die Software muß auf die Bildinformationen, die auf dieser Karte in digitaler Form zwischengespeichert werden, Zugriff haben. Das Programm untersucht die

---

[2] **O**bjekt**r**egistrierung mit **Vi**deokamera und **Co**mputer

[3] **Ca**mera **r**esolved **m**otion **en**coding

Informationen nach dem Überschreiten einer Helligkeitsschwelle und speichert die Koordinaten. Wird die Schwelle wieder unterschritten, so wird der nächste Wert gespeichert. Nach dem Untersuchen aller Bildpunkte ist also die Kontur des hellen Fleckes bekannt. Damit kann der Mittelpunkt dieser Kontur bestimmt und als Position des Objektes gespeichert werden.

Carmen benötigt eine spezielle Digitalisierungskarte, die inzwischen nicht mehr hergestellt wird. Dr. Axel Kuhn, der Entwickler des Systems, ist jedoch daran interessiert, das Programm an eine neuere Karte anzupassen. Nach eigenen Angaben würde der Hersteller diese neue Karte mit den speziellen Zugriffsmöglichkeiten jedoch auch nur einen begrenzten Zeitraum produzieren.

Dieser Nachteil tritt vermutlich immer auf, wenn man sich auf Spezialhardware stützt.

1.3 Analyse von Digitalvideos

Bei folgenden Programmen besteht das Problem, nach wenigen Jahren nicht mehr lauffähig zu sein, nicht. Bei ihnen wird die Bewegung als digitaler Videoclip im Computer gespeichert und anschließend analysiert. Die erforderliche Hardware - eine der vielen Digitalisierungskarten des Marktes - ist mittlerweile standardisiert.

Es gibt inzwischen eine Reihe von Programmen, die sich mit der Analyse von zuvor erzeugten Digitalvideos beschäftigen.

Deutschsprachige Programme sind u. a. [2]:

- **Galileo** (Erhältlich im Internet
  http://www.uni-potsdam.de/u/physik/teach1.htm)
- **Diva (Augsb.)** (Erhältlich bei der Physikdidaktik, Universität Augsburg)
- **Diva (München)** (Erhältlich bei der Physikdidaktik, LMU München)

Bei den Programmen wird die Position eines Objektes bei jedem Bild durch einen Mausklick auf das Videobild erfaßt. Anschließend ist eine graphische Darstellung und Auswertung möglich.

2 Einführung in die Grundlagen der Videodigitalisierung

In diesem Kapitel sollen die Grundbegriffe der Computer - Videotechnik dargestellt werden.

So wie die Digitalisierung von Ton und Musik große Verbreitung gefunden hat, steht es wohl auch der Videodigitalisierung bevor. Die zeitliche Verzögerung gegenüber der Verbreitung von Musik - CD's begründet sich in der noch größeren Datenmenge, die bei Bild + Ton - Informationen anfallen. Diese setzt hochwertige und damit teure Technik voraus, die sich bisher hauptsächlich Studios bei den Fernsehanstalten leisten konnten. Durch die Weiterentwicklung und den Preisverfall dieser Technik wird sie jedoch inzwischen für den Privatanwender erschwinglich. Beispiele sind die Video - CD's, Videokonferenzen, Videosequenzen auf Heimcomputern etc..

Die Qualität digitaler Videos hat inzwischen ein durchaus ansprechendes Niveau erhalten. Die Wiedergabe von Videosequenzen soll nach [15] in Zukunft stärker durch die Grafikkarte unterstützt werden. Damit wird die Darstellungsqualität weiterhin deutlich steigen.

Trotzdem verspricht gute analoge Technik derzeit noch ein besseres Abbild der Wirklichkeit, da stufenlos Veränderungen aufgezeichnet werden können.

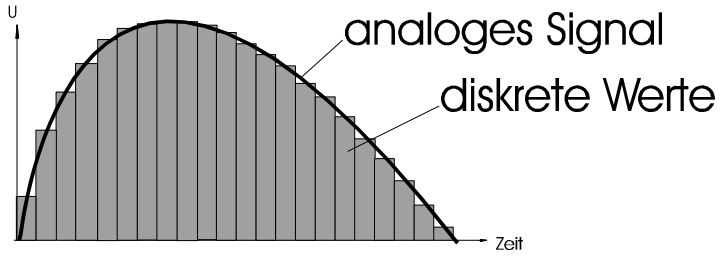


Abbildung 1: Analoge - digitale Signale

Beim Digitalisieren wird dem Signal in äquidistanten Zeitabständen ein Zahlenwert zugeordnet (Abbildung 1). Vorteile gegenüber dem analogen Film bestehen u.a. im qualitätsverlustfreien Kopieren und Archivieren und der einfachen Bearbeitung digitaler Daten.

Um ein analoges Videosignal (von einem Camcorder, einem Videorecorder oder einem Fernsehtuner) digitalisieren zu können - was auch als ‘capturen’ oder ‘grabben’ bezeichnet wird -, benötigt man
a) geeignete Hardware (Videogerät, schneller PC, Digitalisierungskarte) und
b) Software.
Mit der Hardware setzt sich der zweite Abschnitt dieses Kapitels auseinander. Welche Software notwendig ist, wird im dritten Teil beschrieben.
Im folgendem Abschnitt findet eine kurze Einführung in häufig benutzte Grundbegriffe der Videotechnik statt.

2.1 Technische Grundbegriffe

Bei der Darstellung von Videos auf einem Computermonitor ergeben sich einige technische Schwierigkeiten. Ein Computermonitor ist ein wesentlich hochwertigeres Wiedergabemedium als ein Fernseher.

	Videobild (PAL[4])	**Computerbild**
Auflösung	768 x 576 sichtbar	z.Zt. bis 1600 x 1200
Bildwiederholfrequenz	50 Hz (interlaced)	60 bis 120 Hz (non interlaced)
Steuerungssignal	FBAS	RGB

Tabelle 1[3]

2.1.1 Die Videotechnik und ihre

[4] **P**hase **A**lternation **L**ine

Normen

2.1.1.1 Auflösung

Das erste Merkmal in Tabelle 1 wird durch die europäische Fernsehnorm CCIR[5] festgelegt. Die Anzahl der Zeilen beim Bildaufbau beträgt 625 (PAL und SECAM1[6], das in Amerika vorherrschende NTSC[7] verwendet 525 Zeilen bei einer Bildwiederholfrequenz von 60 Hz)[4]. Mit dem zugrundeliegendem Seitenverhältnis 4:3 ergibt sich die horizontale Auflösung 833. Von den 625 Zeilen werden aber nur 576 angezeigt, der Rest wird heute für Videotextinformationen und ähnliches genutzt.

Bei dem zukünftigen Fernsehsystem HDTV[8] soll mit einer sehr hohen Auflösung von ungefähr 1100 Zeilen bei 1920 Pixel pro Zeile (Seitenverhältnis 16:9) gearbeitet werden.

2.1.1.2 Bildwiederholfrequenzen

Als nächste Angabe in Tabelle 1 werden die Bildwiederholfrequenzen verglichen. Durch den schnellen Wechsel einzelner Bilder werden dem menschlichen Auge Bewegungen vorgetäuscht. Diese werden als kontinuierlicher Ablauf interpretiert, sobald der Bildwechsel öfter als 16 mal in der Sekunde erfolgt [5]. Doch selbst bei 25 Hz - der Bildwechselfrequenz unseres Fernsehers - entsteht noch der Eindruck, daß das Bild flimmert. Aus diesem Grund wurde das Zeilensprungverfahren (Interlace-Darstellung) erfunden, welches sich die Trägheit des menschlichen Auges zunutze macht. Dabei wird ein Bild so in zwei Halbbilder zerlegt, daß alle ungeraden und alle geraden Zeilen je ein Halbbild erzeugen (Abbildung 2). Diese werden nun mit 50 Hz - der Frequenz unseres Stromversorgungsnetztes - übertragen, so daß scheinbar

---

[5] **C**omission **C**onsultative **I**nternationale de **R**adiocommunications

[6] **S**équentielle **C**oleur **à M**émoire

[7] **N**ational **T**elevision **S**ystem **C**ommitee

[8] **H**igh **D**efinition **T**ele**v**ision

ein 50-maliger Bildwechsel pro Sekunde entsteht. Die Bildwechselfrequenz für ein Vollbild beträgt aber immer noch 25 Hz.

Abbildung 2: Interlace

Ein Standardmonitor hat dagegen aus ergonomischen Gründen eine viel höhere Bildwiederholfrequenz und arbeitet im Non-Interlaced-Modus.

2.1.1.3 Videosignale

Das dritte Vergleichskriterium in Tabelle 1 ist das Signal, mit dem die Bildinformationen übertragen werden. Computermonitore werden von der Grafikkarte durch RGB - Signale angesteuert. Wie Young[9] und Helmholtz[10] erkannten, lassen sich mit den drei Grundfarben **R**ot - **G**rün - **B**lau nach der additiven Farbmischung alle anderen Farben darstellen. Technisch wird auf drei getrennten Leitungen der jeweilige Pegel des Farbanteils übermittelt. Eine vierte Signalleitung überträgt Synchronisationsinformationen der Bildfrequenz. Bei der Übertragung von Signalen im herkömmlichen Videobereich wäre dieses Verfahren zu aufwendig. Die Bildsignale werden hier über nur eine Leitung übertragen. Außerdem nimmt man Rücksicht auf Schwarz-Weiß-Fernseher, bei denen nur Helligkeitsinformationen angezeigt werden können.
Nach dem Erzeugen eines RGB - Signals in einer Kamera wird dieses also in Helligkeitswerte (*Luminanz*) und Farbwerte (*Chrominanz*) umgewandelt.
Luminanz (Y) betrifft die physiologische Leuchtdichte gemessen in $cd \cdot m^{-2}$. Dieser Helligkeitswert ergibt sich zwingend aus den einzelnen Anteilen von Rot, Grün und Blau. Man kann diesen Zusammenhang in eine Formel fassen:

---

[9]Thomas Young, *1773, brit. Naturwissenschaftler

Y = 0,299 R + 0,587 G + 0,144 B

Da Grün ein heller Farbton ist, muß er am höchsten gewichtet werden, Blau ist relativ dunkel und der Helligkeitswert von Rot liegt dazwischen.

Man übermittelt das Luminanzsignal vollständig, also braucht man bei der Chrominanz (C) nicht mehr alle drei Farbinformationen, eine wäre redundant. Sie läßt sich beim Empfänger nach obiger Formel bestimmen.

2.1.1.4 FBAS, YC, YUV

Das am weitesten verbreitete Signal heißt FBAS (auch Composite - Signal). Hierbei werden nacheinander die **F**arb-, **B**ild-, **A**ustast- und **S**ynchronsignale übermittelt. Man findet es bei Videorecordern und Camcordern, die nach den Standards VHS und Video8 arbeiten.

Die Zusammenlegung der Luminanz und Chrominanz auf eine Leitung bringt einige Schwächen in Bezug auf Kontrast und Farbsättigung mit sich.

Bei den neueren S - VHS und Hi8 Videogeräten findet man das verbesserte Y/C-Signal, bei dem Luminanz und Chrominanz getrennt übertragen werden.

Im professionellen Bereich (auch bei unserer Digitalisierungskarte Miro DC30) findet man das YUV - Signal, bei dem die Chrominanz in zwei Signale aufgespalten wird, also insgesamt drei getrennte Signale verarbeitet werden. Üblicherweise werden bei der Digitalisierung die Farbinformationen, also U und V, nicht so hoch codiert wie die Helligkeitsinformationen Y, da man so die erzeugten Daten ohne große Qualitätseinbußen komprimieren kann.

2.1.2 Kompression

Wollte man die analogen Signale von einem Camcorder oder einem Videogerät in True Color, das sind 24 Bit Farbtiefe (2^{24} = 16,7 Mio. Farben), digitalisieren und unkomprimiert auf einer Festplatte speichern , hätte man das Problem, daß

---

[10]Hermann Helmholtz, *1821, dt. Physiker und Physiologe

das System die immense Datenmenge pro Sekunde nicht verarbeiten könnte und die Festplatte innerhalb kürzester Zeit voll wäre.

Rechenbeispiel:

Bei der Digitalisierung eines Pal - Vollbildes (768x576 Bildpunkte) mit je 24 Bit = 3 Byte fallen für jedes Bild

768 Pixel/Linie · 576 Linien · 3 Byte/Pixel = **1,27** Megabyte an.

Bei 25 Vollbildern pro Sekunde erhält man

1296 KB/Bild · 25 Bilder/s = **31,6** Megabyte pro Sekunde!

Bei der Bewältigung der enormen Datenmengen orientiert man sich an der Natur. Menschliche Sinnesorgane nehmen große Datenmengen auf, weit mehr als das Gehirn tatsächlich verarbeiten kann. Allein an visuellen Informationen erreichen das Auge pro Sekunde 800 Megabyte. Diese werden bei der Weiterleitung an das Gehirn um etwa den Faktor 100 reduziert [5]. Dabei werden Informationen, die nicht entscheidend zum Sinneseindruck beitragen, unterdrückt.

Die audiovisuellen Kompressionsverfahren untersuchen die Daten auch nach unwichtigeren Informationen..

Es gibt grundsätzlich zwei verschiedene Möglichkeiten für eine Kompression:

- Softwarekompression:

  Die Kompressions- und Dekompressionsalgorithmen liegen als Programme vor. Vorteile sind, daß diese Methode kostengünstig ist und sich das erstellte Video auf anderen Computern, die entsprechende Dekomprimierungstreiber installiert haben, wieder abspielen läßt. Bei dem Betriebssystem Win95 sind alle gängigen Videotreiber bereits integriert. Eine hochwertige Softwarekompression ist jedoch sehr rechenintensiv. Da die gesamte Komprimierung und Dekomprimierung von dem Prozessor übernommen wird, ist diese Möglichkeit relativ leistungsschwach.

  Viele Kompressionsalgorithmen arbeiten asymmetrisch, d.h. sie benötigen beim Komprimieren mehr Zeit als beim Dekomprimieren. Das Abspielen eines Videos in voller Auflösung und 25 Bildern pro Sekunde auf Software-

basis ist schon möglich. Für die Komprimierung ist das in naher Zukunft nicht zu erwarten.

- Hardwarekompression:
  Die Komprimierung durch spezielle Chips auf der Digitalisierungskarte ist wesentlich schneller. Gängige Videokarten sind heute in der Lage, Full - Screen- (Erfassen aller Pixel) und Full - Motion - Aufnahmen (Erfassen aller Bilder) plus Audiodigitalisierung in S - VHS Qualität durchzuführen.

Weiterhin kann man zwischen verlustfreien (lossless) und verlustreichen (lossy) Kompressionsmethoden unterscheiden.

Lossless - Kompression:

Hier wird versucht, die Informationen zusammenzufassen. Alle Informationen sind anschließend noch enthalten.

Beispiele :

1. RLE[11] -Verfahren:
   Aus *Weiß-Weiß-Weiß-Weiß-Weiß-Schwarz* wird *5xWeiß-Schwarz*.
2. Lempel - Ziv - Verfahren : Dieser Algorithmus durchsucht die Originaldaten nach gleichen Bytefolgen. Bei der Codierung des Satzes "Kers*ting* grüßt Schlich*ting"* würde das zweite *ting* nur durch einen Verweis auf das erste gespeichert werden.
3. Huffmann - Verfahren : Hierbei durchsucht man ein Bild auf die Häufigkeit der verwendeten Farben. Die am häufigsten auftretende Farbe bekommt den kürzesten Code, die am zweithäufigsten auftretende den zweitkürzesten Code usw. (Ähnlichkeit zum Morsen : das e als häufig auftretender Buchstabe bekommt hier eine kurze Tonfolge.)

Je nach Video lassen sich so Kompressionsfaktoren zwischen 1:2 und 1:3 erreichen. Die Datenmengen sind damit aber immer noch zu groß um von handelsüblichen PC’s verarbeitet werden zu können.

Lossy - Kompressoren :

Diese Kompressoren eliminieren weniger wichtige Informationen aus den Filmen. Je besser ein Kompressor zwischen bildtragenden und vom Auge nicht wahrnehmbaren Elementen unterscheiden kann, desto besser wird die Bildqualität.

In der vorliegenden Arbeit wurden die Lossy - Verfahren YUV 4:2:2 (Kap.2.1.2.1.) und M-JPEG (Kap.2.1.2.2.) benutzt.

2.1.2.1 YUV 4:2:2

Bei dieser Datenreduktion werden die Farbinformationen U und V nur halb so oft gespeichert wie die Helligkeitsinformationen Y. Also werden anstatt 8:8:8 Bit = 24 Bit pro Pixel nur 8:4:4 Bit =16 Bit pro Pixel benötigt. Der Faktor liegt bei 3:2, und das obwohl das Signal nach [5] immer noch volle Studioqualität besitzt. Das Verhältnis 4:2:2 gilt als optimal für das menschliche Auge. Ältere Digitalisierungskarten arbeiten mit einer Reduktion auf 4:1:1, die Folge sind jedoch sichtbare Verluste bei Farbqualität und -Brillanz [6].

2.1.2.2 Motion - JPEG

Dies ist das meistverbreitetste Verfahren zum Online - Digitalisieren von Videos. Zur Echtzeitkomprimierung und zum Abspielen von M-JPEG[12]-codierten Filmen benötigt man eine Hardwareerweiterung.

JPEG ist ein Verfahren zur Komprimierung einzelner Bitmap-Grafiken. Der zugrundeliegende Algorithmus heißt Diskrete-Cosinus-Transformation (DCT). Dabei wird ein Bild in Blöcke von üblicherweise 8x8 Pixel unterteilt. Jeder Block wird dann in einen Frequenzbereich umgewandelt, wobei hohe Frequenzwerte unberücksichtigt bleiben. Anschließend findet noch eine verlustfreie Reduzierung mit einer Packmethode statt, meistens mit dem Huff-

---

[11] **R**un **L**ength **E**ncoding

mann - Verfahren. Bei den häufig auftretenden schwachen Farb- und Helligkeitsveränderungen in einem Block entstehen nur wenige Werte [7]. Der Kompressionsfaktor läßt sich dabei beliebig einstellen. Möglich sind Werte bis über 100:1. Bis zu einem Faktor von 15:1 finden dabei nach [5] keine sichtbaren Einbußen in der Qualität statt. Diese Angabe bezieht sich aber auf die Betrachtung eines Films. Bei schnell wechselnden Bildern ist eine hohe Qualität jedes einzelnen nicht entscheidend, wohl aber bei der Analyse von Videosequenzen (siehe dazu auch Kap. 3.5.3.).

Motion-JPEG ist nichts anderes als das Aneinanderreihen von JPEG-komprimierten Bildern. Leider ist das Verfahren nicht standardisiert, so daß auf einem Computer mit M-JPEG- Hardwareerweiterung die durch eine andere Digitalisierungskarte erstellten Filme möglicherweise nicht gelesen werden können, sie müssen dann erst in ein anderes Format umgerechnet werden (siehe Software - CODECs).

2.1.2.3 MPEG

Die MPEG[13] - Kompression gilt als zukunftsweisend auf dem Gebiet Video und Computer (z.B. benutzen Video - CD's zur Wiedergabe von Spielfilmen und CDI-Player diesen Standard).

Bei der Kompression werden zuerst M-JPEG - Bilder errechnet (Intraframecodierung). Anschließend wird nach Ähnlichkeiten zwischen Bildern in einer Videosequenz gesucht und nur die Veränderungen gespeichert (Interframecodierung). Aus den Einzelbildern werden sogenannte P-, B- und I-Frames[14] erzeugt:

---

[12]**M**otion **J**oint **P**hotographic **E**xperts **G**roup

[13]**M**otion **P**icture **E**xperts **G**roup

[14]Der Begriff 'Frame' (übersetzt Rahmen, Bild) wird von der Literatur unterschiedlich benutzt. In ihm werden Bild und Ton eines Zeitintervalles gespeichert. Ich benutze den gängigen Begriff gleichbedeutend mit Einzelbild, da ich keinen Ton digitalisiere.

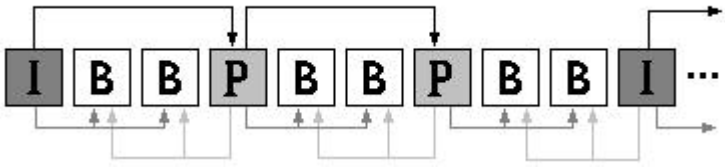


Abbildung 3: MPEG - Frames

- **Intra-Frames (I)** sind die Schlüsselbilder mit einer vollständigen Bildinformation.
- **Predictive-Frames (P)** enthalten nur die Veränderungen zum vorangegangenen I- oder P - Frame. Dabei wird nicht nur pixelweise verglichen, sondern es wird auch nach ähnlichen Bildbereichen gesucht, und beim Auffinden solcher nur der Versatz vektoriell gespeichert (z.B. bei Kameraschwenks). Bei der Dekodierung wird immer auch das vorangegangene I - Frame benötigt.
- **Bidirectional-predictive-Frames (B)** versuchen, einzelne Bildpunktänderungen vorherzusagen. Bei einem bewegten Objekt braucht man so nur den Anfangs- und Endpunkt zu speichern, die Zwischenpositionen lassen sich bei der Dekompresssion interpolieren. Der Algorithmus nimmt das vorhergehende und das nachfolgende P- oder I- Frame und tastet sich gewissermaßen in die Mitte vor.

Es werden tatsächlich viel mehr B- Frames geschaffen als in Abbildung 3 gezeigt. Pro Sekunde wird meist nur ein einziges vollständiges I- Frame gespeichert (einstellbar). Damit erreicht man Kompressionsfaktoren bis 1:200 bei guter Qualität [4].

MPEG - Videos benötigen zu der aufwendigen Dekodierung entweder eine Hardwareerweiterung (moderne Grafikkarten besitzen diese MPEG - Beschleuniger) oder eine Abspielsoftware wie z.B. ‘XING’ für Videos in kleiner Auflösung (Win95 beinhaltet in der Betaversion ‘active movie’ als MPEG- Player). Das Digitalisieren von Videosequenzen in Echtzeit ist nicht praktikabel, für eine Sekunde sind einige Minuten Rechenzeit von Hochlei-

stungsrechnern nötig. Es existieren zwar auch Hardwareunterstützungen, diese sind für den Privatanwender z.Zt. jedoch unerschwinglich.
CD-Roms sind derzeit das einzig attraktive Medium für digitale Videos. Da die erreichbaren Übertragungsraten dabei aber begrenzt sind, legt der MPEG - Standard fest, daß der Datenstrom eine festgelegte Grenze nicht überschreitet:
MPEG I : 183 KByte/s - das entspricht dem Datenstrom, den die ersten CD-Rom-Laufwerke übertragen können (MPEG II : 1,2 MByte/s - 8xCD-Rom-Laufwerk, MPEG III ist in Planung, MPEG IV berücksichtigt niedrige Raten als Basis für Videokonferenzen über ISDN) [3].

MPEG - codierte Videosequenzen sind zur physikalischen Analyse nicht geeignet, da die
- B - Frames nicht den gefilmten Ort eines Objektes wiedergeben, die Position wurde interpoliert.
- Digitalisierung langwierig ist.
- Bearbeitung unter Windows schwierig ist, Einzelbilder sind nicht ohne weiteres anwählbar.

2.1.2.4 Fraktale Kompression

Dieses noch nicht sehr verbreitete Verfahren versucht, die im Bild auftretenden Formen mathematisch zu beschreiben. Dabei wird innerhalb von Einzelbildern nach immer wieder auftretenden Mustern gesucht. Die gründliche Analyse zum Auffinden von redundanten Elementen, die anderen ähnlich sehen und daher beim Speichern überflüssig sind, dauert natürlich sehr lange (einige Sekunden pro Einzelbild), es können nach [4] aber Kompressionsraten von 10000:1 erreicht werden! Die Qualität ist auflösungsunabhängig, mit den Formeln läßt sich das Bild in der jeweiligen Auflösung berechnen.

2.1.2.5 Software - Codecs[15]

Auch wenn eine Hardwareunterstützung (M-JPEG) zur Digitalisierung fast unabdingbar ist, möchte man die Videos auch auf anderen PC's abspielen. Dazu werden die zuvor digitalisierten M-JPEG - Videos in ein anderes Format übersetzt, das dann ohne eine Erweiterungskarte wiedergegeben werden kann. Das Umrechnen braucht nicht mehr in Echtzeit zu erfolgen und kann je nach Codec mehrere Minuten für einige Sekunden dauern.

Das übliche Format im PC - Bereich unter Windows heißt **A**udio **V**ideo **I**nterleave und unter der Endung *.avi* werden die Dateien auch gespeichert. Es gibt jedoch Unterschiede in den *avi* - Formaten und so läßt sich nicht jeder Film dieser Endung unter Windows abspielen. Voraussetzung ist, daß der Codec, mit dem der Film erstellt wurde, auch als Treiber installiert ist. Die üblichen Codecs sind Video1, RLE, Cinepak und Indeo (s. Abbildung 4, zu erreichen unter Start→Systemsteuerung →Multimedia). Ab dem Betriebssystem Win95 sind die Codecs in dem Betriebssystem integriert, für ältere Windowsversionen werden sie im Internet kostenlos angeboten, z.B. unter http://members.aol.com/raacc/wim.html

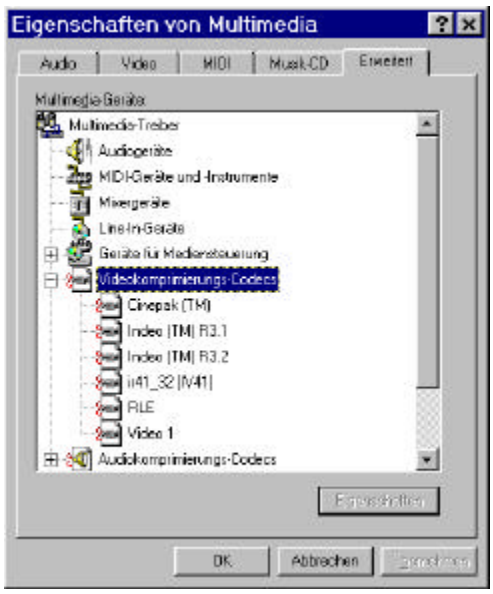


Abbildung 4: Videotreiber

Die Codecs unterscheiden sich in wichtigen Einzelheiten:

- RLE Dieser ältere Softwarekompressor arbeitet nach der Run - Lenght - Encoding - Methode (siehe Lossless - Kompression). Die Farbtiefe beträgt jedoch nur 8 Bit.
- Video 1 ist ebenfalls schon älter und arbeitet auch mit einer Farbreduktion auf üblicherweise 8 (max. 16) Bit. Er ist zum Einsatz auf alten Rechnern sinnvoll, bei denen

[15] **Co**mpressor/**Dec**rompressor

geringe Datenraten bei passablen Ergebnissen gewünscht werden.

- Cinepak läßt Kompression in True Color (24 Bit Farbtiefe) zu. Die Bildqualität ist hochwertig. Es benötigt zum Komprimieren wesentlich mehr Rechenzeit (ab Faktor 60) als zur Wiedergabe (asymmetrisches Verfahren). Die Kompressionsrate liegt bei ca. 10:1.
- Indeo Dieser ausgereifte Codec wird nach [3] früher oder später alle anderen Codecs verdrängen. Moderne Grafikkarten unterstützen die Wiedergabe. Als reine Softwarelösung ist nach [4] die Qualität jedoch schlechter als Cinepak

Quicktime von Apple (.mov) auf Macintoshrechnern ist das Analogon zu .avi unter Windows. Inzwischen ist auch ein Übersetzer für Windows erhältlich, so daß auf die große Mac - Video - Datenbank zugegriffen werden kann: http://www.public.iastate.edu/%7Estark/gutil_sv.html#Windows

2.2 Hardware

Um ein Videosignal digitalisieren zu können, benötigt man neben einem geeigneten Computer noch eine Zusatzkarte.
Diese Karten werden im zweiten Abschnitt vorgestellt. Welche Anforderungen an den Computer gestellt werden, beschreibt der folgende Abschnitt.

2.2.1 Systemanforderungen

Für die Digitalisierung von Videosequenzen sollte ein PC folgende Minimalanforderungen erfüllen:

- DX2 - Prozessor mit 66 MHz
- 16 MB RAM Arbeitsspeicher
- 700 MB Festplatte

Für moderne Karten (wie zum Beispiel die während dieser Arbeit benutzte ‘Miro DC30’) empfiehlt der Hersteller folgende Ausstattung:

- Pentium PC mit 100 MHz PCI
- Win 95 Betriebssystem
- 24 MB RAM Arbeitsspeicher
- SCSI oder E-IDE Festplatten mit hohen Datenraten, Speicher > 1GB
- Monitor mit großer Bildfläche
- eine hochwertige Grafikkarte mit mindestens 600 x 800 Pixel in True Color

Bei der Analyse bereits existierender Computervideos ergeben sich bis auf längere Wartezeiten keine Probleme durch langsame Rechner. Nur die Grafikkarte und der Monitor sollten auch hierbei obigen Ansprüchen genügen.

2.2.1.1 Monitor und Grafikkarte

Zur Analyse digitaler Videosequenzen sollten auf dem Bildschirm folgende Fenster sichtbar sein:

1. Das aktuelle Videobild
2. Einige Bedienelemente des Programms
3. Die Anzeige des Suchergebnisses

Bei einer Auflösung von 600 x 800 Pixel kann es zu Problemen bei der Anzeige aller Fenster kommen. Besser ist eine Auflösung von 1024 x 768 in True Color. Falls jedoch in diesem Modus die Farbtiefe nicht mehr möglich ist, empfielt es sich, die Einstellung 600 x 800 Pixel beizubehalten.

2.2.1.2 Festplatte

Die Geschwindigkeit der Festplatte ist ein wesentlicher Punkt für eine erfolgreiche Digitalisierung.

Der M-JPEG - Kompressionsfaktor kann so angepaßt werden, daß das System die Daten gerade noch verarbeiten kann. Dabei sollte aber ein gewisser Sicherheitsspielraum gelassen werden: Die Datentransferrate der Festplatte sollte ca.

um 1/3 höher sein als die erzeugte Datenrate der Videokarte [8]. Wird diese Leistungsreserve zu klein gewählt, so können kleine Störungen das System überlasten und einen ‘Datenstau’ verursachen. Die Bilddaten können nicht rechtzeitig gespeichert werden, bevor 1/25 s später die Daten für das nächste Bild geliefert werden.

In so einem Fall läßt die Digitalisierungssoftware ein Frame einfach ‘unter den Tisch’ fallen (lost frame), gibt aber eine entsprechende Meldung aus. Bei der Betrachtung der Videos fällt es nicht auf, ob eins von 25 Bildern pro Sekunde fehlt, zur Videoanalyse ist es jedoch verheerend. Durch das Fehlen eines Bildes entsteht bei der Auswertung ein unerklärlicher Sprung des bewegten Objektes auf.

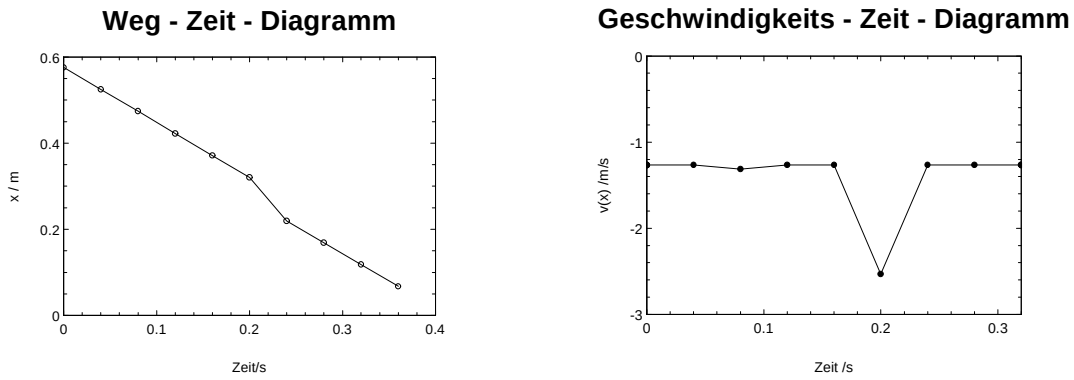


Abbildung 5

Gründe für diese kleinen Störungen können Betriebssystemabfragen, ausgelöste Interrupts, Repositionierung der Leseköpfe von Festplatten etc. sein [3].

Um eine Festplatte zu testen, ist im Lieferumfang der Miro DC30 das Programm EXPERT enthalten. Es schreibt eine Datei von 20 MB auf die Festplatte und liest diese wieder ein. Dabei mißt es die Schreib- und Lesegeschwindigkeiten. Bei dem Lehrstuhlrechner entstand folgendes Geschwindigkeitsprofil:

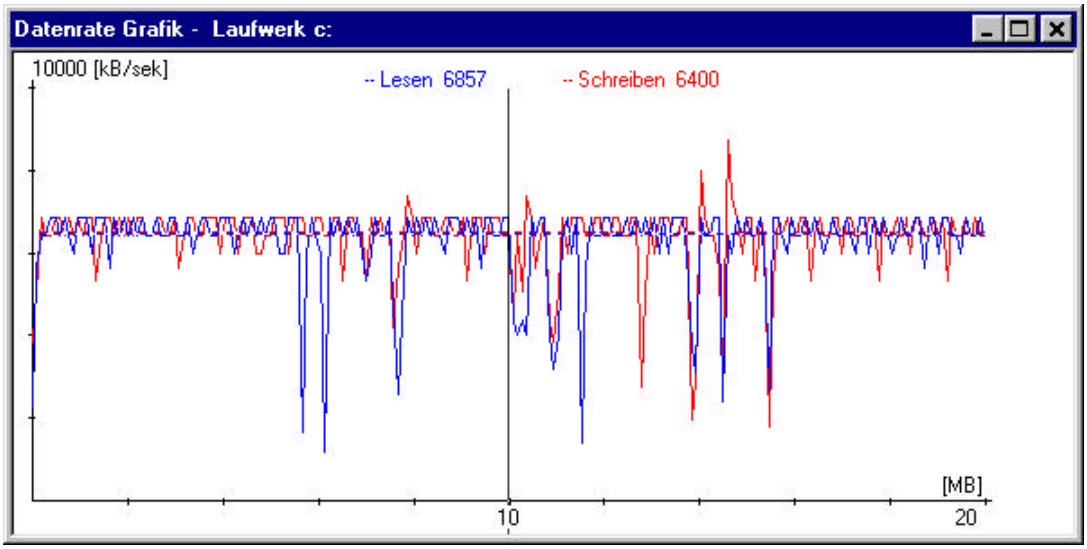


Abbildung 6: Schreib- und Lesegeschwindigkeit der Festplatte

Zu erkennen sind deutliche Schwankungen sowohl beim Lesen als auch beim Schreiben. Die Schreibgeschwindigkeit liegt durchschnittlich bei 6400 KB/s . Das Programm schlägt zur Digitalisierung jedoch nur eine Datenrate von 5011 KB/s vor, damit während eines Geschwindigkeitseinbruches kein Bild verlorengeht. Diese Datenrate reicht für unsere Zwecke völlig aus.

Die Digitalisierungskarte Miro DC30 kann bei minimaler Kompression 6 MB/s liefern. Diese Datenraten leisten bisher nur SCSI- oder schnelle E-IDE Festplatten. Miro bietet im Internet einen Informationsdienst mit Testergebnissen aller gängigen Festplatten an. Die Massenspeicher wurden speziell auf ihre Eignung zur Videodigitalisierung untersucht:

http://www.miro.de/d/d2-produkte/d24-digitalvideo/dc30comp_neud.html

Die Herstellerangaben der Festplatten - Datenrate sind oft durch einen Platten-Cache geschönt [9]. Zur Videodigitalisierung ist dieser Cache (üblich ca. 128 KB) auf jeden Fall zu klein.

Bei dem Abspielen von Videos am PC kennt man, besonders bei langsameren Rechnern, das Problem von nicht fließender, ruckelnder Wiedergabe. Dies liegt in der Regel nicht an verlorenen Frames bei der Digitalisierung, sondern an einem zu langsamen Rechner beim Abspielen. Dieser zeigt dann einige Bilder des Videos nicht an, um einen Zeitlupeneffekt zu verhindern. Eine Videowie-

dergabe mit 240 x 180 Pixel in 256 Farben schafft heute fast jeder Rechner problemlos.

2.2.2 Digitalisierungskarten

Der größte Käuferkreis von Digitalisierungskarten[16][17] sind Hobbyfilmer. Durch das Digitalisieren bieten sich komfortable Bearbeitungsmöglichkeiten der privaten Aufnahmen (Betitelung, Schneiden mit Überblendungen, Nachvertonung und etliche Spezialeffekte). Die Karten werden so zum Massenprodukt und damit billig. Nach dem Bearbeiten werden die Videos meistens wieder auf Kassette zurückgespielt.

Eine andere Anwendung ist das Erstellen von Videos für Computeranwender. Bei älteren Karten ließen sich oft nur Videos in der Größe 160 x 120 Pixel erstellen, da die M-JPEG - Funktion fehlte. Eine andere Möglichkeit war die Einzelbilddigitalisierung: Dabei stellt man einen Videorecorder auf Pause und digitalisiert das Standbild. Dann spielt man immer ein Bild weiter und faßt zum Schluß alle Bilder zusammen. Die Standbilddarstellung des Recorders ist jedoch meist nicht störungsfrei.

Inzwischen sind Karten durch M-JPEG - Encoderchips in der Lage, das volle PAL - Format mit 25 Bildern pro Sekunde in Echtzeit zu erfassen. Der Kompressionsfaktor von M-JPEG ist dabei so gering (z.B. 1:3), daß sich keine sichtbaren Einbußen zwischen dem analogen Originalfilm und dem digitalen Film auf dem Monitor bemerkbar machen (laut Herstellerangaben volle S-VHS - Qualität).

Heutige Karten arbeiten fast alle mit der *Overlaytechnologie*, um das eingehende Videosignal direkt auf dem Monitor sichtbar zu machen. Das Signal braucht zur bloßen Ansicht nicht digitalisiert zu werden, denn das würde das System zu sehr belasten. Die Grafikkarte läßt den Bereich, in dem das Video

---

[16]Vorsicht bei dem Begriff Videokarte, manche Hersteller bezeichnen ihre Grafikkarten mit Videobeschleunigern genauso.

[17]In der Literatur werden auch die Begriffe *Frame-Grabber* oder *Capture-Boards* benutzt.

erscheinen soll, schwarz. In diesen Bereich schreibt die Overlaykarte nun direkt über das Bussystem die Videobildinformationen im RGB - Format - ohne den Prozessor zu belasten. Die Videodaten kann man so noch nicht analysieren, für den Computer ist an der Stelle nur ein schwarzes Fenster. Somit ist eine direkte Onlineanalyse bewegter Objekte nicht möglich, die Videodaten müssen sich erst im Speicher befinden.
Die Installation der Karte ist völlig problemlos. Lediglich die Kabelverbindungen zum Videorecorder waren unvollständig (Adapter Scart auf Chinch).
Um die Digitalisierungskarte steuern zu können, benötigt man spezielle Programme (s. Kap. 2.3.).

2.3 Digitalisierungssoftware

Bei der Erstellung digitaler Videosequenzen hat die Software folgende Aufgaben:

Steuern der Digitalisierung
↓
Schneiden/Kürzen der Sequenzen
↓
Speichern im gewünschten Format

Einige Programme haben sich als Standardsoftware entwickelt, da sie mit vielen Digitalisierungskarten mitgeliefert werden.
Dies sind:

- **Video for Windows** (Microsoft): Das Programmpaket brachte damals den Einstieg in den Umgang von Video am PC, einem Bereich, der bis dahin von Macintosh - Rechnern dominiert wurde. Die Runtime - Version (Codecs) befinden sich heute auf fast jedem PC (s. Kap. 2.1.2.5.).
- **Premiere** (Adobe) : Eine Komplettlösung zum Digitalisieren, Bearbeiten (div. Filter), Schneiden (Überblendeffekte), Titelgenerator etc.

- **Mediastudio** (Ulead) : Ein Programm nach dem Vorbild Premiere, jedoch deutlich günstiger und mit einigen Zusatzfunktionen (z.B. Morph - Editor).

Daneben gibt es eine Reihe anderer Programme wie ‘Splice’ oder ‘Granit’, die man günstig erwerben kann (z.B. auf beiliegender Buch - CD von [6]). Diese bieten jedoch nicht den Komfort und die Möglichkeiten wie Premiere oder Mediastudio [11]. Eines dieser beiden Programme gehört inzwischen grundsätzlich zum Lieferumfang moderner Videokarten [10].

Bei uns war es Premiere 4.2 LE (Light Edition) und Video for Windows. Die meisten Funktionen sind dabei zur Video- und Audio- Nachbearbeitung konzipiert - zum Aufbereiten von Heimvideos. Diese Funktionen benötigt man für unsere Ziele nicht.

Nach einheitlicher Meinung der Fachliteratur unterscheidet sich die Bedienung der verschiedenen Programme nur unwesentlich.

3 Die Aufnahme von Videosequenzen

Ob eine Videosequenz für eine automatische Analyse geeignet ist, entscheidet sich bereits bei der Aufnahme. Damit die Position eines Objektes durch das in Kapitel 4 entwickelte Programm verfolgt werden kann, muß sich das Objekt deutlich von seiner Umgebung unterscheiden. Durch welche Maßnahmen und Einstellungen das ermöglicht wird und worauf noch zu achten ist, soll in diesem Kapitel beschrieben werden.

Der erste Abschnitt beschäftigt sich mit dem Aufbau und der Beleuchtung des Versuches. Der zweite Abschnitt zeigt, welche Objektmarkierungen geeignet sind. Wie die Kamera bei den besonderen Anforderungen einzustellen ist, liest man in Abschnitt 3. Im vierten Abschnitt werden die Einstellungen der Digitalisierungssoftware behandelt.

3.1 Versuchsaufbau

Die Bewegung des Objektes muß in einer Ebene parallel zur Bildebene der Kamera verlaufen, da dreidimensionale Bewegungen nicht verwertbar sind. Entfernt sich das Objekt von der Kamera oder nähert es sich an diese an, so würde dies eine scheinbare Positionsänderung bewirken (siehe Abb.7).

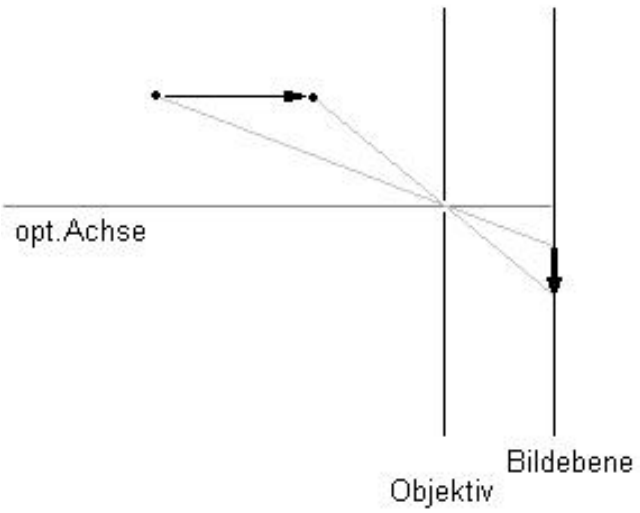


Abb.7: Dreidimensionale Bewegung

In der Regel wird man der Einfachheit halber von einem ruhendem Bezugssystem und damit einer ruhenden Kamera ausgehen wollen. Da Verwackelungen große Ungenauigkeiten bei der späteren Analyse mit sich bringen, ist ein Kamerastativ zwingend notwendig

Hintergrund:

Nach Herstellerangaben steuert die Digitalisierungskarte die Kompression so, daß jedes Bild eine bestimmte Datenmenge erhält. Die Festplatte wird also immer mit der konstanten Datenrate beschrieben. Ist der Bildinhalt detailarm, müssen weniger Bilddaten komprimiert werden, es gehen also nicht so viele Informationen verloren.

Daraus folgert sich für uns, daß der Hintergrund und die Umgebung möglichst einfach gestaltet werden sollten, damit mehr Daten zur Beschreibung der Markierung übrigbleiben.

Dies deckt sich mit der Forderung, daß nicht zu viele Farben im Bild auftauchen sollten, um zufällige Übereinstimmungen mit der Markierungsfarbe zu vermeiden.

Als problematisch erweisen sich blanke Metallgegenstände. In ihnen können sich Gegenstände (zum Beispiel bunte Kleidungsstücke von Schülern) widerspiegeln.[18] Durch blanke Stativstangen traten bisher jedoch noch keine Störungen auf.

Maßstab:

Um bei der späteren Analyse einen Eichfaktor zu finden, welche Wegstrecke einem Pixel entspricht, muß in dem Bild ein Objekt bekannter Länge sichtbar sein. Dazu kann man ein Maßband zu dem Versuchsaufbau legen oder ein Objekt ausmessen. Die Strecke sollte sich aber über einen möglichst großen Bildausschnitt erstrecken. Ist sie nur wenige Pixel groß, so läßt sich der Eichfaktor nur mit großem Fehler bestimmen.

Beleuchtung:

[18]Ich wunderte mich beim Filmen von grünen Markierungen über einige grüne Pixel auf einer blanken Muffe. Schuld waren die Lüftungsaufbauten auf dem Dach hinter den Laboratorien.

Bei der Ausleuchtung des Versuches ist zu beachten, daß die Helligkeit der Farbmarkierungen an verschiedenen Positionen nicht zu sehr schwanken darf, um die spätere Analyse nicht zu erschweren. Dies erreicht man, indem die Lichtquelle direkt neben oder über der Kamera steht (s. Abb. 8), so daß sich der Abstand zwischen Lichtquelle und Objekt während der Bewegung nicht ändert. Falls dabei jedoch störende Reflexionen auftreten, muß man die optimale Stellung suchen, möglicherweise sogar eine zweite Lampe aufstellen. Es ist aber zu beachten, daß kein Schattenwurf auf der Markierung stattfinden sollte.

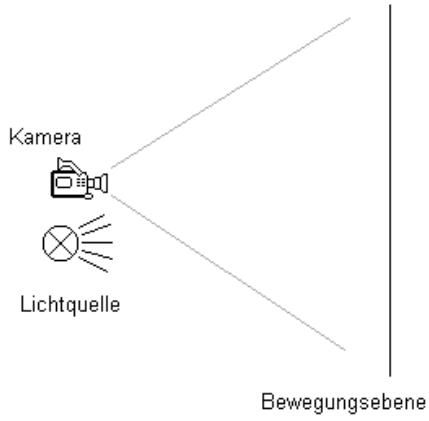


Abb. 8: Position der Lichtquelle

Diffuses Licht ist grundsätzlich besser zur Beleuchtung geeignet.

Bei längeren Aufnahmen sollte kontrolliert werden, daß sich die Beleuchtungsstärke zwischendurch nicht zu stark geändert hat (z.B. Wolke vor der Sonne).

Bei extrem kleinen Verschlußzeiten muß für starke Beleuchtung gesorgt werden (s.Kap. 3.4.2.).

3.2 Farbmarkierungen

Zur Bestimmung einer Position muß entweder das Objekt selbst erkannt oder eine Markierung angebracht werden. Da das Erkennen einer Objektform äußerst schwierig und unzuverlässig ist [12], sucht der Algorithmus in dem Programm **Viana** ausschließlich nach bestimmten Farben (s.u.). Die Farbmarkierungen müssen jedoch gewisse Eigenschaften besitzen.

3.2.1 Größe und Form

An den Grenzen der Markierungen kommt es durch die Quantisierung in Pixel immer zu relativ unvorhersehbaren Mischfarben (siehe Abb. 9). Wichtig ist nun, daß in der Mitte der Markierung noch einige Pixel die Originalfarbe besitzen (Feld 2), die in jedem Folgebild zu finden ist. Die Markierung muß dazu eine minimale Größe besitzen.

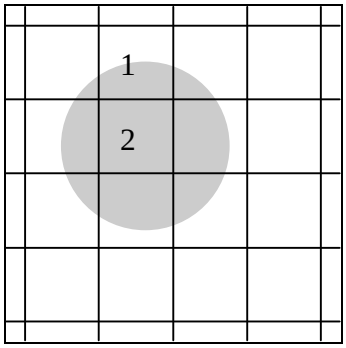


Abb. 9: Quantisierung

Erfahrungswert: Unter sieben Pixel Durchmesser können schon Probleme entstehen.

Bei der Verpackungsindustrie (z.B. unter Milchpackungen) oder bei Crash-Test-Dummies finden sich oft mehrfarbige Markierungen (z.B. Abb. 10, oben). Da es hier aber wieder zu viele Farbgrenzen gibt - schließlich kommt es auch an den Grenzen zwischen den beiden Farben zu Farbmischungen -, ist diese Markierungsart unter einer bestimmten Größe für uns nicht geeignet.

Auch andere Formen wie Rechtecke, Dreiecke, Buchstaben etc. müssen relativ groß sein, um als solche erkannt zu werden.

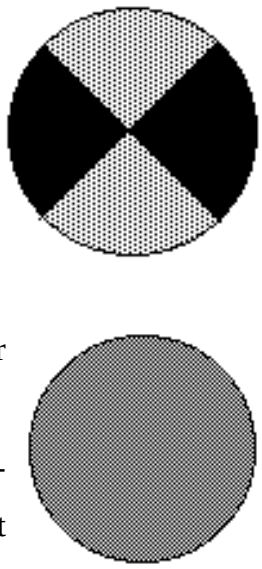


Abb. 10: Markierungen

Auch wenn unterschiedliche Formen neben der Farbe ein zusätzliches Argument liefern könnten, ob es sich bei den gefundenen Bildpunkten nun um das Objekt oder um Störungen handelt, wurden ausschließlich runde Markierungen benutzt (Abb. 10, unten). Bei jeder anderen Form wäre die Grenzlänge größer. Außerdem verschiebt sich der Mittelpunkt bei einer Rotation der Markierung nicht (wie z.B. bei unsymmetrischen Buchstaben).

3.2.2 Farbe

Eine ähnliche Farbe wie die der Markierung sollte möglichst nicht im restlichen Bildbereich vorkommen. Man muß sich also zuerst die Umgebung des bewegten Objektes anschauen, um dann eine geeignete Markierungsfarbe zu wählen.

Beispiel: Findet der Aufbau eines Versuches auf einem braunen Holztisch statt, so ist die Farbe Rot nicht sehr geeignet, da sich Braun und Rot farblich zu sehr gleichen.

Die Idee, Schwarzlichtfarben zu benutzen, die bei UV - Bestrahlung stark leuchten, mußte leider verworfen werden. Um eine gleichmäßige Beleuchtung über die ganze Bewegungsebene zu erzielen, sind mehrere Schwarzlichtröhren nötig. Mit nur einer schwachen Lampe muß man diese sehr nah an das Objekt aufstellen. Dies hat jedoch zur Folge, daß die Markierung an verschiedenen Orten einmal zu stark und das andere mal zu schwach leuchtet.

3.2.3 Material

Markiert man das zu verfolgende Objekt mit flüssiger Farbe, so sollte man ausschließlich matte Farbe verwenden. Bei glänzenden Farben kann es bei einzelnen Bildern durch Reflexion zu Farbverschiebungen kommen.

Auch bei ausgeschnittenen Papiermarkierungen ist auf eine matte Oberfläche zu achten. Geeignet scheint hier Tonpappe.

3.3 Kameraeinstellungen

Die Einstellmöglichkeiten verschiedener Kameras unterscheiden sich je nach Preisklasse. Die in diesem Kapitel beschriebenen Grundeinstellung sind jedoch schon bei einfachen Kameras möglich. Auch wenn ich mich hier auf den Kameratyp: *Sony CCD TR707E PAL* beziehe, dürfte eine Übertragung auf andere Kameras nicht schwer fallen.

3.3.1 Zoom

Je nachdem, wie weit die Kamera von dem bewegten Objekt aufgestellt wird, muß man den Bildausschnitt mit der Zoomeinstellung anpassen. Die Bewegung sollte über den ganzen Bildbereich verlaufen, um so eine maximale örtliche Auflösung zu erzielen. Ein kleiner Sicherheitsabstand vom Rand ist jedoch sinnvoll, da hier manchmal Signal - Übertragungsfehler und damit unvorhersehbare Farben auftauchen können.

Grundsätzlich zu vermeiden sind Extremeinstellungen des Zooms:

Bei großem Weitwinkel treten Tonnenverzerrungen auf (s. Abb. 11). Abb. 13 zeigt die Aufnahme eines karierten Blattes. Der Abstand zum Objektiv der Kamera beträgt lediglich 25 cm. An den Rändern des Bildes kann man die leichte Wölbung nach außen erkennen.

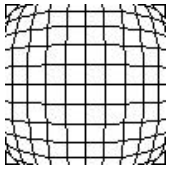


Abb. 11

Beim maximalem Zoom zeigt sich statt dessen eine Kissenverzerrung (s. Abb. 12). Abb. 14 zeigt das karierte Blatt jetzt in einer Entfernung von 1,5 m. Diesmal läßt sich eine - wenn auch sehr schwache - Wölbung nach innen beobachten.

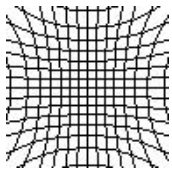


Abb. 12

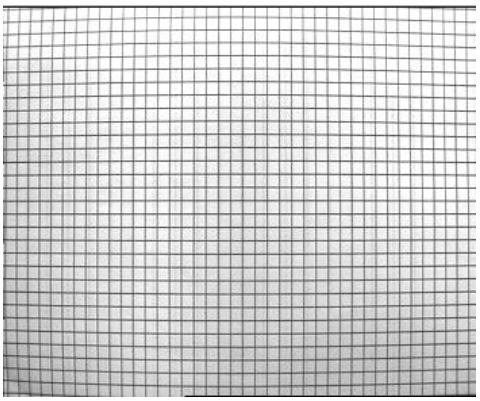


Abb. 13: maximaler Weitwinkel

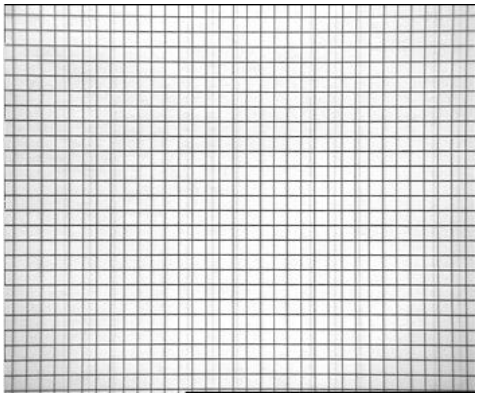


Abb. 14: maximaler Zoom

In einem Zoombereich von ca. 20% bis 90% ist keine Verzerrung erkennbar.

3.3.2 Verschlußzeiten[19]

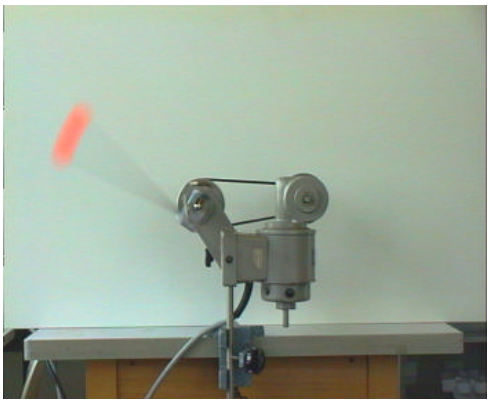
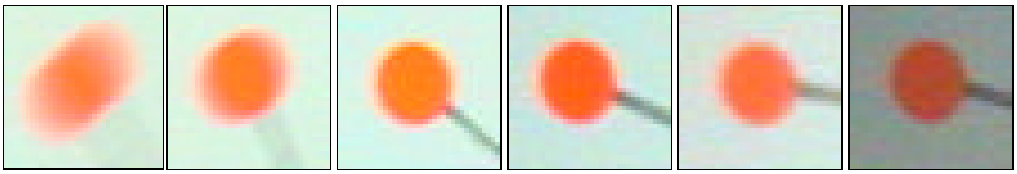


Abbildung 15: Rotation

Filmt man ein bewegtes Objekt mit automatischer Verschlußzeit - Einstellung (Grundeinstellung beim Einschalten der Kamera), so erhält man unscharfe Einzelbilder. Abbildung 15 zeigt eine schnell[20] rotierende Markierung ($\omega = 18s^{-1}$, v = 4m/s). Während der Aufnahme des Einzelbildes bewegte sich die Markierung um ca. 15° weiter (errechnete Verschlußzeit: ca. 1/70 s).

An der Kamera gibt es mit der Einstellung ‘Shutter’ die Möglichkeit, 6 kürzere Zeiten zu wählen:



1/125 s 1/250 s 1/1000 s 1/2000 s 1/4000 s 1/10.000 s

Abbildung 16: Verschlußzeiten

Bei 1/125 s Verschlußzeit erkennt man in der Mitte der Markierung die Originalfarbe, in Bewegungsrichtung findet am Rand jedoch eine Vermischung mit der Hintergrundfarbe statt (hier wegen des hellen Hintergrundes eine Farbverblassung). Dies führt bei der späteren Analyse zu Problemen, da man das Bild nach einer bestimmten Farbe untersuchen will.

---

[19] es müßte eigentlich Öffnungszeiten (der Blende) heißen

[20] mit schnell ist hier nicht die Absolutgeschwindigkeit gemeint, sondern die Geschwindigkeit des Objektes relativ zur Bildschirmhöhe (Pixel pro Sekunde).

Wie man in Abbildung 16 sieht, ist die Markierung ab 1/1000 s scharf (selbst bei der 7- fachen Vergrößerung des Bildes). Solch hohe Geschwindigkeiten des Objektes wie in diesem Beispiel werden eher die Ausnahme sein. Mit einer Einstellung von 1/1000 s ist eine scharfe Darstellung damit gewährleistet. Bei sehr kleinen Verschlußzeiten (siehe 1/10.000 s) muß für zusätzliche Beleuchtung gesorgt werden.

3.3.3 Belichtung

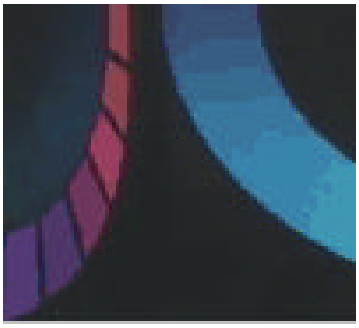


Abb.17: Autoexposure

Filmt man in Innenräumen, ist das Bild oft sehr dunkel (siehe Abb.17). Die Farben wirken nicht so strahlend wie bei der Originalvorlage. An der Sony - Kamera gibt es die Möglichkeit, nach Drücken des Schalters ‘Exposure’ die automatische Belichtung auszustellen und die Helligkeit manuell zu verändern. Bei zwei bis drei Stellungen über dem Normalwert sind die Farben satter (Abb. 18), auch wenn das Bild an hellen Stellen jetzt etwas überbelichtet ist. Die Farben lassen sich bei der Bildanalyse wesentlich besser erkennen. Beim Filmen unter freiem Himmel ist das Erhöhen der Belichtung nicht nötig.

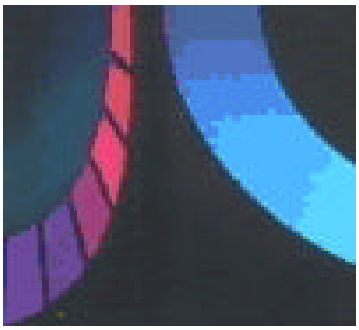


Abb. 18: Exposure +2

Grundsätzlich sollte man beim Filmen immer die Automatik ausstellen, damit sich nicht mitten in der Sequenz die Belichtungseinstellung ändert. Diese Änderung vollzieht sich nicht stufenlos, und so kann sich der Farbwert der Markierung beim zufälligen Aktivieren der Automatik von einem zum anderen Bild drastisch ändern.

3.3.4 Autofokus

Die Kameraautomatik stellt immer die Gegenstandsweite des Bildmittelpunktes scharf dar. Bewegt sich unser Objekt in diesem Gebiet, so

paßt die Automatik die Fokussierung an. Da die optimale Fokussierung nicht sofort erreicht, sondern erst ein paar Bilder lang gesucht wird, führt dies zu einigen unscharfen Einzelbildern. Der Autofokus sollte also ausgestellt werden (Schalter ‘Focus’).

3.3.5 Weißabgleich

Verschiedene Lichtquellen haben unterschiedliche Farbtemperaturen. Das Licht der Mittagssonne (hohe Farbtemperatur) hat einen hohen Blauanteil, das Licht einer Glühlampe (niedrige Farbtemperatur) dagegen einen hohen Rotanteil. Moderne Kameras besitzen einen automatischen Weißabgleich, damit z.B. ein weißes Blatt Papier unter verschiedenen Beleuchtungssituationen auch weiß erscheint.

Jedoch kann auch diese Automatik während des Filmens eine Veränderung der Einstellungen bewirken, so daß sich die Farben des Filmes von einem zum anderen Bild verändern. Um das zu verhindern, stellt man auch den automatischen Weißabgleich besser aus (Schalter ‘Wht Bal’). Die Beleuchtung darf sich während des Filmens natürlich nicht mehr ändern.

3.3.6 Zusammenfassung

Folgende Reihenfolge kann nun vor der Aufnahme empfohlen werden:

1. Aufstellen der Kamera, der Beleuchtung und des Versuchsaufbaus
2. Einstellen des Zoomfaktors, so daß das Objekt nicht aus dem sichtbaren Bereich wandern kann
3. Einstellen der Verschlußzeit (Shutter)
4. Eventuell Erhöhen der Belichtung (Exposure)
5. Ausstellen des Autofokus
6. Ausstellen des automatischen Weißabgleichs

3.4 Softwareeinstellungen

Nachdem der Versuch aufgebaut und die Kamera eingestellt wurde, kann die Digitalisierung beginnen. Jedoch sind auch hierbei noch einige Optimierungen möglich.

3.4.1 Bildrate

Bei schnellen Bewegungen (z.B. freier Fall) sollte man die maximale Bildrate einstellen, also 25 Bilder/s. Damit erreicht man eine optimale zeitliche Auflösung. Bei langsamen Bewegungen (z.B. Pohl'sches Rad) sind nicht so viele Bilder notwendig, die Position ändert sich vielleicht von einem zum anderen Bild gar nicht (nur unterhalb der Meßgenauigkeit). Eine hohe Anzahl an Bildern bringt dann keine qualitative Verbesserung der Auswertung.
Bei der Angabe der Bildrate sollten nur Werte eingestellt werden, welche die Bildrate der Kamera (25 B/s) ganzzahlig teilen. Wählt man beispielsweise 10 B/s, so stimmen Bild und Zeitinformationen nicht mehr überein (Abbildung 19):

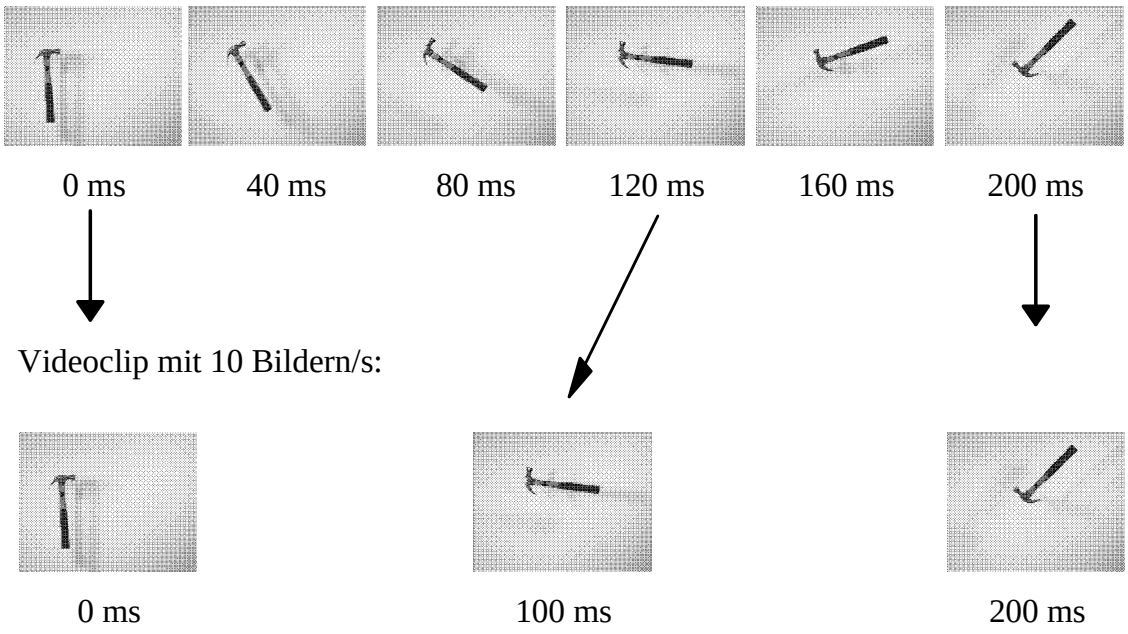


Abbildung 19: Zeitverschiebung

Bei einer Bildrate von 10 Bildern pro Sekunde müßte jedes ‘zweieinhalbte’ Bild des Filmes aufgenommen werden. Da aber kein Bild zum Zeitpunkt 100 ms existiert, wird einfach das nächste Bild digitalisiert. Dieses Bild zeigt das Objekt jedoch zum Zeitpunkt 120 ms. Die Position des Hammers ist in Abbildung 19 also schon ein Stück weiter als in der Realität zum Zeitpunkt 100 ms. Bei 200 ms stimmen Zeit- und Bildinformationen wieder überein. Damit ist jede zweite Bildinformation zeitlich leicht verschoben, und es darf bei der Analyse nicht mehr von konstanten Zeitintervallen zwischen den Bildern ausgegangen werden.
Sinnvolle Bildraten sind also 1, 5 oder 25 Bilder pro Sekunde.

3.4.2 Farbsättigung

In diesem Abschnitt soll gezeigt werden, daß sich durch das Erhöhen der Sättigungseinstellung eine farbige Markierung besser von dem Hintergrund unterscheiden läßt.

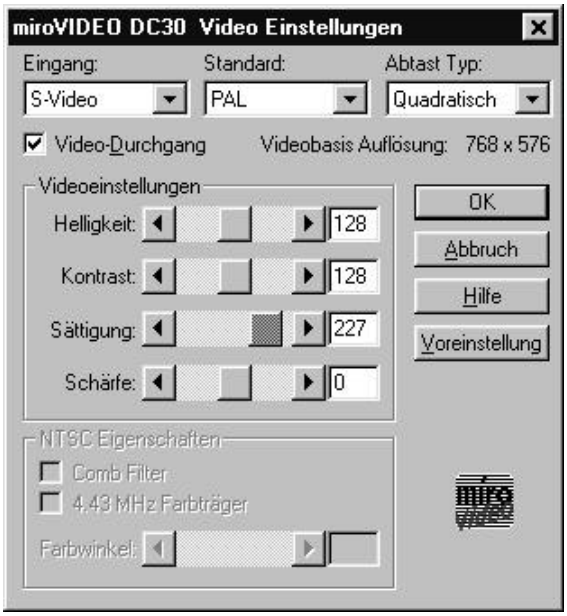


Abbildung 20

Digitalisiert man eine sehr leuchtende Farbvorlage, so erscheint diese in dem erzeugten Computerfilm blasser und mit einem leichten Graustich (Abb. 21). Sie hebt sich bei einigen Videosequenzen somit nicht ausreichend vom Hintergrund ab. Dies läßt sich nur zum Teil durch das Erhöhen der Belichtung kompensieren (s. Kap. 3.4.3.).
Bei den Grundeinstellungen der Digitalisierungskarte findet sich aber neben Helligkeit, Kontrast und Schärfe eine Möglichkeit, die Farbsättigung einzustellen (Abbildung 20).

Abb. 21: Normalstellung

Abb. 21 zeigt eine digitalisierte rote Markierung bei der Grundeinstellung (mittlere Sättigung). Die Markierung besteht aus ca. 10.000 Pixel.

Zur weiteren Untersuchung der Markierungsfarbe soll ihr Spektrum angezeigt werden. Dazu werden die Pixel der Markierung (ohne grauen Randbereich) ausgelesen und in ihre R,G,B - Werte getrennt. Nun trägt man die Anzahl der Pixel mit einem bestimmten Farbwert gegen alle möglichen Farbwerte (0 - 255) auf. Dadurch entstehen drei Diagramme, die in Abbildung 22 dar-gestellt wer-den.

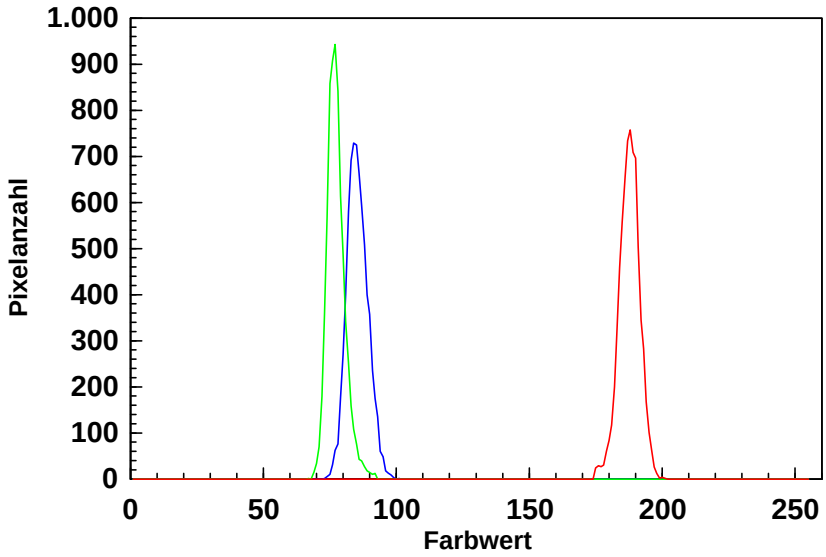


Abbildung 22: Normalstellung der Sättigung

Die meisten Pixel dieser Markierung haben einen Rot - Wert von 188, es treten aber auch andere R - Werte (zwischen 173 - 198) auf.

Ähnliche Aussagen lassen sich auch über die Grün - und Blau - Werte treffen. Uns soll es im Augenblick genügen, nur die Maximalwerte zu erfassen. Für die Markierung wären das R:188, G:80, B:85.

Die Hintergrundfarbe eines Versuchsaufbaus wird in der Regel relativ farblos sein. Dies bedeutet, daß die R,G,B - Werte nah beieinander liegen. So ergeben z.B. für die Werte:

R:0, G:0, B:0 → Schwarz

R:100, G:100, B:100 → Dunkelgrau

R:200, G:200, B:200 → Hellgrau

R:255, G:255, B:255 → Weiß

Je weiter die Maximalwerte unserer Markierungsfarbe auseinander liegen, desto besser lassen sie sich von diesen Grautönen unterscheiden. Der optimale Rotwert wäre R:255, G:0, B:0. Dieser Wert wird jedoch selten zu finden sein. Wird die Markierung nun mit erhöhter Sättigung digitalisiert (Abb.24), so hat man sich dem Ideal 255:0:0 etwas angenähert.

Abb.24:

Sättigung 80%

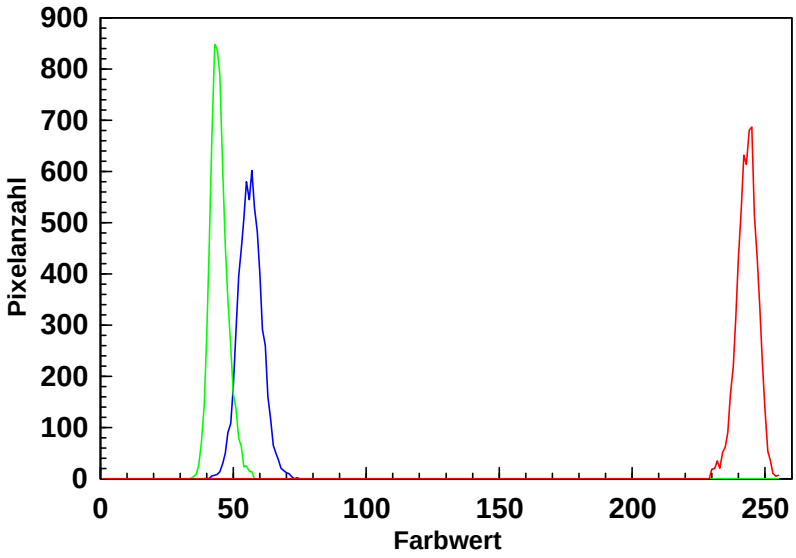


Abbildung 23

Die Peaks liegen
hier bei 245:45:60 (Abbildung 23).
Wesentlich ist nun, daß die Grauwerte des Hintergrundes von der Sättigungserhöhung kaum verändert werden. Bei ihnen liegen die R-,G-,B- Werte noch immer nah beieinander und sie werden nur wenig in die Randgebiete gedrängt.

Stellt man die Sättigung jedoch zu hoch ein, so können auch Brauntöne zu Rot werden, während sich die Werte der Markierungsfarbe kaum noch verbessern (Abb. 26). Das Diagramm in Abbildung 25 zeigt, wie sich die Rotwerte auf 255 und 254 häufen (Maßstabsänderung beachten). Für die Videoanalyse bringt das keine entscheidenden Vorteile.

Abb. 26: Max. Sättigung

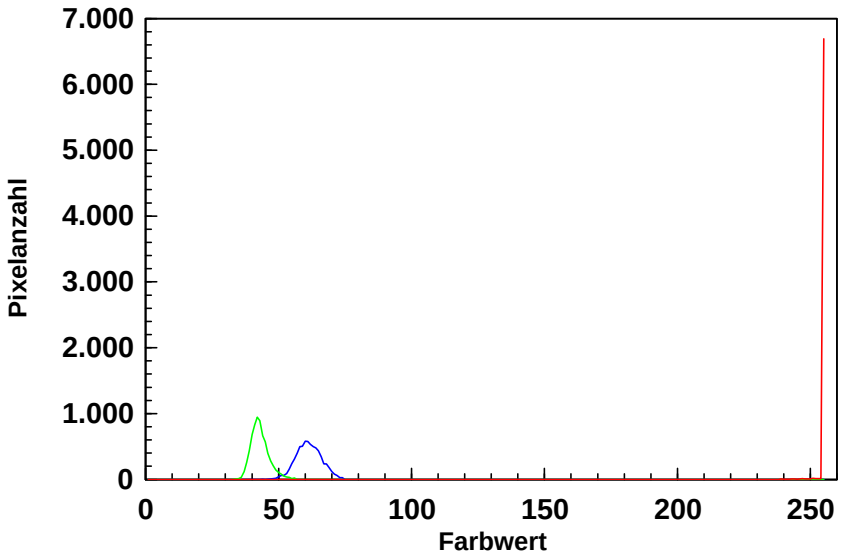


Abbildung 25

3.4.3 M-JPEG- Kompressionsfaktor

Vergleicht man die Qualität zweier Videos, von denen einer mit einem M-JPEG - Faktor von 3:1 und der andere mit 27:1 digitalisiert wurde, findet man beim Abspielen kaum Unterschiede, wohl aber bei der Einzelbildbetrachtung. Um das zu zeigen, wurde eine relativ kleine blaue Markierung gefilmt (Abb. 27) und mit unterschiedlichen Kompressionsfaktoren digitalisiert. Anschließend wurde das Bild nach dem Blauton durchsucht. Abbildung 28 zeigt die 6- fachen Vergrößerungen des blauen Kreises und rechts daneben die Suchergebnisse. Pixel mit einem ähnlichen Farbton wie der Mittelpunkt wurden schwarz gekennzeichnet.

Abb. 27

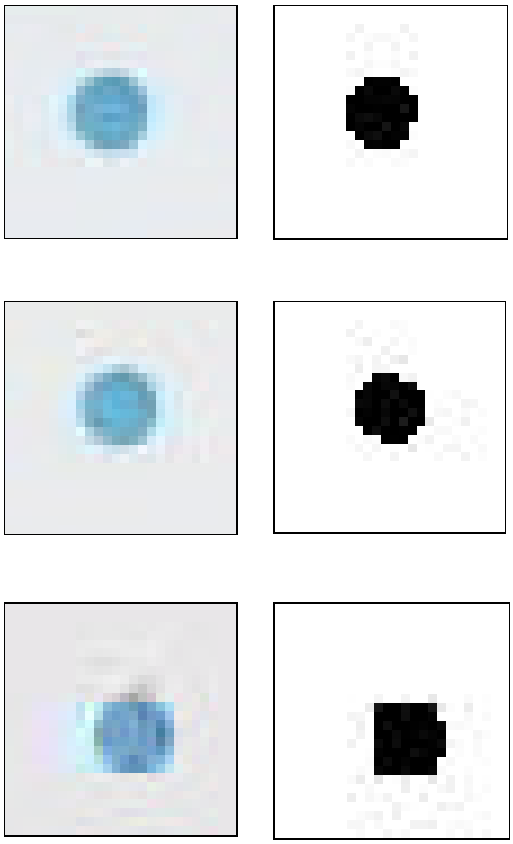


Abbildung 28

Bei der ersten Vergrößerung (Kompressionsfaktor 3:1) in Abbildung 28 ist das Blau relativ einheitlich, nur am Rand treten Verfälschungen auf (siehe Kap. 3.3.1.).

Darunter (Faktor 12:1) sind die Blautöne der einzelnen Pixel schon unterschiedlicher. Man muß also nach einem größeren Blaubereich suchen (genaue Vorgehensweise s. 4.1.3.). Dies kann zur Folge haben, daß auch andere Pixel des Bildes, die nicht zur Markierung gehören, aber zufällig auch in diesen Blaubereich fallen, erkannt werden. Bei der letzten Vergrößerung (Faktor 27:1) zeigt sich die Klötzchenbildung durch die Unterteilung des Bildes in 8x8 Pixel große Bereiche. Der erkannte Mittelpunkt kann damit ein bis zwei Pixel neben der tatsächlichen Position liegen. Solch hohe Kompressionsfaktoren sind jedoch nur bei sehr langsamen Rechnern nötig.

Zur Verdeutlichung der M-JPEG - Fehler wurde eine bewußt kleine Markierung und eine unruhige Umgebung gewählt (siehe Kap. 3.1.).

3.4.4 Systemeinstellungen

Werden während des Digitalisierens *lost frames* angezeigt, so kann das an den Grundeinstellungen des Systems liegen. Folgende Maßnahmen tragen zur Vermeidung dieses Problemes bei:

1. Der Festplattenbereich, in den die Daten geschrieben werden, sollte nicht zerstückelt, sondern zusammenhängend sein. Beim Springen des Schreibkopfes stauen sich sonst die Daten. Bewerkstelligen läßt sich das mit dem Hilfsprogramm ‘Defrag’.

2. Die Übertragungsrate einer Festplatte ist abhängig von der Geschwindigkeit v zwischen Schreibkopf und Magnetscheibe. In den äußeren Spuren der Platte ist diese natürlich größer als in den inneren ($v = \omega \cdot r$, ω= konst.). Festplatten werden von außen nach innen beschrieben, sie sind gegen Ende ihrer Speicherkapazität also langsamer. Um den schnellen Anfangsbereich zu nutzen, ist es also sinnvoll, entweder ein zweites, leeres Laufwerk zur Digitalisierung zu nutzen oder eine Festplatte zu partitionieren. Dies ist bei der Formatierung möglich, man unterteilt sie in beliebiger Größe und gibt den beiden Teilen z.B. die Namen c: und d:. Sie werden fortan wie zwei getrennte Laufwerke angesehen. Das erste läßt man frei und nutzt es nur zum Capture - Vorgang. Damit entfällt auch das häufige Defragmentieren.
3. Nach [10] ist bei Win95 auch die Autostart-Funktion des CD-Rom-Laufwerkes für Unterbrechungen verantwortlich. Diese läßt sich ausstellen unter Start→Einstellungen→Systemsteuerung→System→Gerätemanager→CD-Rom→Automatisches Benachrichtigen
4. Die Uhr in der Win95 Task - Leiste am unteren Bildschirmrand sollte unter Start→Eistellungen→Taskleiste deaktiviert werden, ebenso andere Einstellungen oder Programme, die Interrupts auslösen.

Reichen diese Maßnahmen nicht aus, so muß man den Datenstrom verkleinern durch:

- Vergrößern des Kompressionsfaktors
- Verkleinern der Auflösung (z.B. 320 x 240 Pixel)
- Verringerung der Farbtiefe, bei M-JPEG - Codierung jedoch nicht möglich
- Herabsetzten der Bildanzahl (z.B. 15 statt 25 Bilder/s)

An dem Lehrstuhlrechner waren diese Maßnahmen nicht notwendig, denn die SCSI - Festplatte ist sehr schnell.
Bei kurzen Videoaufnahmen (z.B. freier Fall) besteht auch die Möglichkeit, in den schnellen Arbeitsspeicher zu speichern (RAM >16 MB).

4 Beschreibung des Programms "Viana"

Ziel des **VI**deo**ANA**lyseprogrammes ist die automatische Positionserfassung eines gefilmten Objektes. Die Anzeige und der Umgang mit Videoclips im DOS - Modus ist sehr umständlich. Einfacher gestaltet sich diese Arbeit auf einer höheren Computerebene wie OS-2, Windows etc. Da Microsoft Windows im PC - Bereich große Verbreitung gefunden hat, stützt sich das Programm auf die Vorteile dieses Betriebssystemes.
Windows bietet dem Programmierer die Möglichkeit, auf gängige Funktionen zurückzugreifen. Die große Anzahl von Einzelfunktionen sind über *Windows - API* ansprechbar. Um unter einer Programmiersprache wie Turbo- Pascal oder Basic die Möglichkeiten dieser nicht sehr benutzerfreudlichen Windows-Unterprogramme nutzen zu können, wäre eine sehr lange Einarbeitungsphase nötig.
Mit dem Erschaffen der objektorientierten Sprachen wie *Visual Basic* oder *Delphi* wurde dieser Mißstand abgeschafft. Der Programmierer muß nur noch in Ausnahmefällen eine Windows- API direkt ansprechen.

Damit das Programm auch in der Schule eingesetzt werden kann, muß es dem Benutzer eine möglichst benutzerfreundliche und verständliche Oberfläche bieten. Außerdem sollten von dem Anwender nicht zu viele Parameterangaben gefordert werden, die ihn verwirren könnten.
In Abschnitt 4.1. werden die wesentlichen Passagen des Programmquelltextes dokumentiert und in 4.2. die Bedienung der Oberfläche vorgestellt .

4.1 Entwicklung

Zur Entwicklung des Programmes wurde die Programmierumgebung Borland Delphi 1.0 verwendet. Viele Sprachelemente ähneln denen der Programmier-sprache Borland Turbo Pascal. Es existierte zwar schon die neuere Version Delphi 3.0 zur Programmierung von 32 - Bit - Anwendungen, diese sind jedoch nur unter Windows 95 oder NT lauffähig. Da aufgrund der

angespannten Finanzlage in vielen Schulen auch in Zukunft noch Computer mit älteren Windowsversionen existieren werden, wurde Delphi 1.0. benutzt. Die Abwärtskompatibilität von Windows gewährleistet, daß **VIANA** mittelfristig auf allen PC´s lauffähig bleiben wird.

Beim Programmieren wurde nach Möglichkeit versucht, den Quelltext unkompliziert zu halten, damit das Programm wart- und erweiterbar ist.

Der Quelltext ist im Anhang A abgedruckt. In diesem Kapitel wird auf bestimmte Zeilen des Anhangs verwiesen. Der Nachteil objektorientierter Programmierung ist, daß dieser Quelltext nicht mehr so einfach zu lesen ist wie zum Beispiel ein Turbo - Pascal - Listing, da kein Hauptprogramm existiert. Der Programmablauf erfolgt nicht mehr linear, also ein Befehl nach dem anderen. Je nach Aktion des Benutzers auf der Bedienoberfläche wird eine der 68 Prozeduren ausgeführt. Welche Prozedur bei welcher Aktion aufgerufen wird, kann mit Hilfe der Projektdatei auf der Diskette verfolgt werden. Wer mit dieser Programmierart vertraut ist, kann auch den Ablauf nachvollziehen. Die Namen der Variablen und Oberflächenobjekte sind weitgehend selbsterklärend. Um sich nicht zu sehr im Programmdetail zu verlieren, sollen in den folgenden 12 Unterpunkten nur die wesentlichen Programmteile und spezielle Prozeduren erläutert werden.

4.1.1 Die Videodarstellung

Das Abspielen von Medien gestaltet sich mit Delphi sehr einfach. Die Komponente *Mediaplayer* bietet dazu viele Möglichkeiten. Mit der Komponente *Opendialog* kann man das Standardfenster zum Ermitteln eines Dateinamens und -pfades erzeugen (s. Zeile 158-162) und diese an den Mediaplayer übergeben (Zeile 177,178). Zur Anzeige der momentanen Bildnummer wurde zusätzlich ein *Timer* eingebaut, der einmal pro Sekunde die Position überprüft und ausgibt (Zeile 501-507). Ein automatisches Abspielen des Videos hat jedoch den Nachteil, daß oftmals einzelne Bilder ausgelassen werden können. Daher wurde noch eine Möglichkeit geschaffen, den Film über eine Schleife Bild für Bild abspielen zu lassen (Zeile 922 - 941 und 1564).

Dies erfolgt bei großen Abmessungen und langsamen Computern mit einem Zeitlupeneffekt.

4.1.2 Auslesen von Farbinformationen

Sämtliche Informationen des Filmes findet man in den Dateien auf der Festplatte. Das Format, in dem die komprimierten Informationen codiert und gespeichert werden läßt sich zwar entschlüsseln, jedoch ist dieses bei allen avi - Codecs (s. Kap. 2.1.2.5.) verschieden. Man müßte für Cinepakfilme, Indeofilme, etc. jeweils eigene Dekompressionsalgorithmen programmieren.
Da diese Algorithmen aber schon im System integriert sind (s. Abbildung 4), liegt es nahe, sie auch zu nutzen. Unser Mediaplayer greift auf die Codecs zurück und zeigt uns die Filminformationen von jedem lesbaren Format auf dem Bildschirm an. Da wir den Film an jeder Stelle anhalten können, lassen sich die Standbilder nun untersuchen. Die Geschwindigkeitsnachteile dieser Methode lassen sich an späterer Stelle wieder etwas ausgleichen. Mit der Funktion *GetPixel(GERÄTEKONTEXT,XKoordinate,YKoordinate)*
kann man die Farbinformation eines gewünschten Pixel auslesen. In unserem Fall ist unser ‘Gerät’ das Videofenster. Zuvor muß ein Informationskanal zu diesem Fenster erzeugt werden (Zeile 222 - 229). Da Windows nur eine begrenzte Anzahl solcher Kanäle vergeben kann, muß genau darauf geachtet werden, sie nach dem Erzeugen (auch beim vorzeitigem Abbruch einer Prozedur) wieder zu schließen (Zeile 231 - 237). Ansonsten können Systemprobleme auftreten.
Die übergebene Farbinformation beinhaltet die Rot, Grün und Blauwerte des Bildpunktes:

00000000111111110000000011111111

unben. Byte3: B Byte2: G Byte1: R

Da man mit diesem Longintegerwert schlecht arbeiten kann, wird er in seine drei Anteile umgerechnet. Jeder R,G,B - Wert liegt also zwischen 0 und 255.

Farbe auf Mausklick:

Wird eine Position des Videofensters angeklickt, so startet automatisch eine Prozedur (Zeile 1608), welche die Koordinaten der Maus übergibt. In den Zeilen 249 - 265 wird an diesen Koordinaten nun der Farbwert ausgelesen und in die Rot- Grün und Blauanteile entschlüsselt. Nachdem der Gerätekontext wieder geschlossen wurde, können in der Farbanzeige (Zeile 267 - 294) die drei Anteile graphisch auf der Bedienoberfläche dargestellt werden.

4.1.3 Die Suche nach bestimmten Farben

Um die Position einer farbigen Markierung zu erfassen, sollten von dieser fast alle Bildpunkte erkannt werden. Nur dann ist eine korrekte Mittelpunktsbestimmung möglich.

Durch einen Mausklick soll jetzt eine Farbe gewählt - und alle Pixel des Bildes mit dieser Farbe erkannt werden. Mittels einer Schleife liest ein Hilfsprogramm dazu die Pixel eines Videobildes ein und vergleicht jedes mit der zuvor gewählten Farbe. Stimmen alle drei Farbanteile überein, so wird die Koordinate markiert.

Als Beispiel dient hier ein gefilmter Farbkreis (Abbildung 30). Innerhalb des weißen Rahmens wurde der Farbton orange gewählt. Abbildung 29 zeigt das Ergebnis des Vergleiches (vergrößerter Ausschnitt).

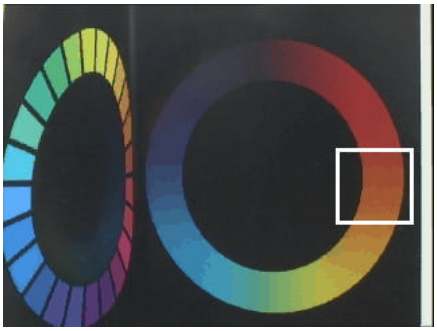


Abbildung 30: Farbkreis

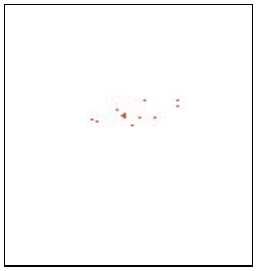


Abbildung 29

Es fanden sich nur 13 Pixel mit den gleichen R,G,B - Werten. Daß nicht mehr Bildpunkte erkannt wurden, liegt daran, daß bei 16,7 Millionen darstellbaren Farben viele Nuancen von Orange existieren, das Programm aber nur nach einer speziellen suchte.

Zur Verdeutlichung des Sachverhaltes wird nun eine andere Darstellungsform gewählt. Trägt man alle 13 erkannten Pixel des Farbwertes gegen den Farbwert (zerlegt in die R,G,B - Anzeile) auf, so erhält man das Diagramm in Abbildung 31 (s. auch Kap. 3.5.2.).

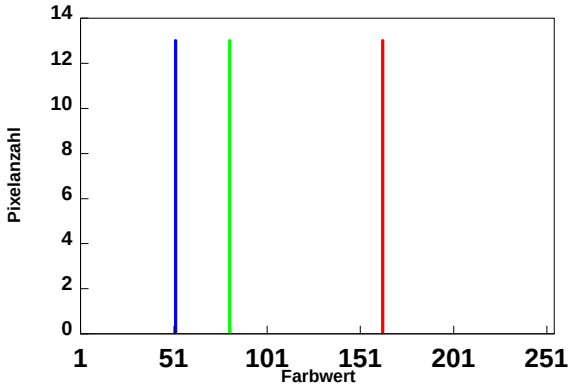


Abbildung 31

Um auch die benachbarten Pixel erkennen zu können, muß nach einem größerem Spektrum gesucht werden.

Im folgenden sollen alle Farben, deren R,G,B - Werte geringfügig von dem gewählten Orange abweichen, noch erkannt werden. In unserem Beispiel hat das Orange die Werte (166, 78, 52). Sollte eine untersuchte Farbe einen Rotwert von 166±11, einen Grünwert von 78±11 und einen Blauwert von 52±11 besitzen, so wird sie markiert. Die erlaubte Abweichung von dem jeweiligen Wert (in diesem Beispiel 11) wird in dem Quelltext '*Toleranz*' genannt. Das Ergebnis dieser Suche zeigt sich in Abbildung 32.

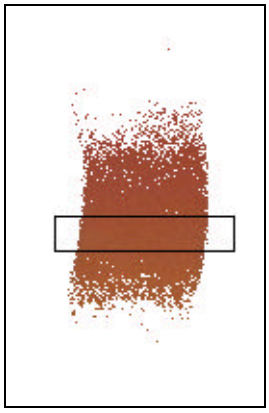


Abbildung 32: Toleranz 11

Innerhalb des schwarzen Rechtecks (nachträglich eingefügt) werden jetzt alle Pixel des Farbkreises erkannt. Außerhalb des Rechtecks weicht die Farbe langsam in den Rot- bzw. Gelbbereich ab.

Die programmtechnische Realisierung dieser Suche findet man in den Zeilen 1043 - 1063.

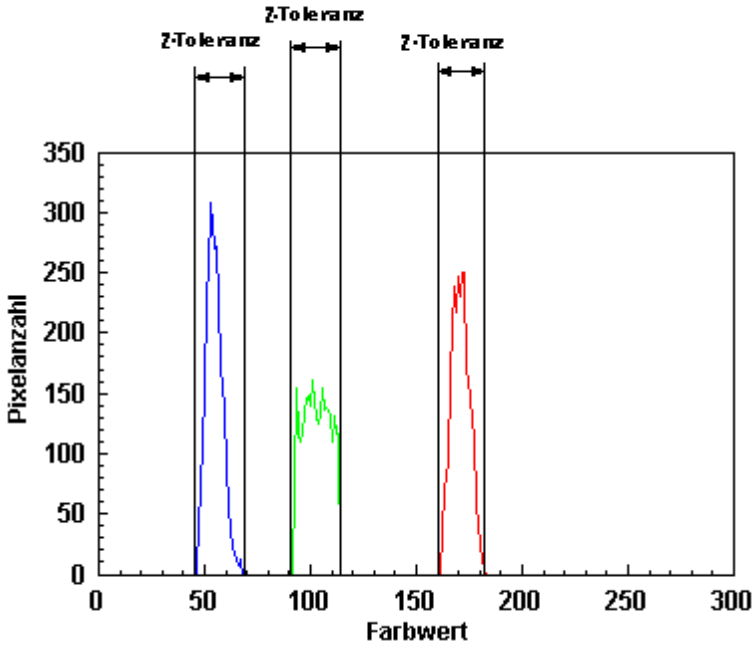


Abbildung 33

Abbildung 33 zeigt alle gefundenen Pixel in der Spektraldarstellung. Nur Bildpunkte innerhalb der Toleranzbereiche wurden markiert.

Prinzipiell wäre für jeden R,G,B - Farbanteil eine eigene Toleranzangabe möglich. Das würde jedoch die Programmbedienung erschweren.

Bei einer großen Toleranz werden auch Rot und Gelbtöne erkannt (Abbildung 34).

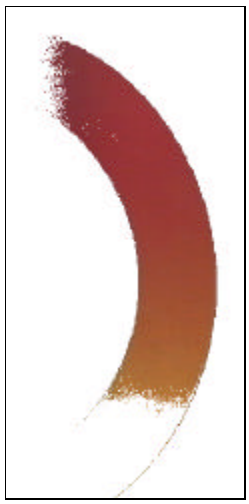


Abbildung 34: Toleranz 40

Bei der Suche nach einer Farbmarkierung kann eine zu große Toleranz den Nachteil mit sich bringen, daß einige Bildpunkte außerhalb der Markierung erkannt werden. Dies führt zu Problemen bei der Mittelpunktsbestimmung (s.u.).

Prinzipiell wären für jeden Farbanteil R,G,B eine eigene Toleranzangabe möglich. Das würde die Programmbedienung jedoch erschweren.

4.1.4 Zweitsuche

Bei einigen Filmen läßt sich der Toleranzbereich nicht zur Zufriedenheit einstellen.

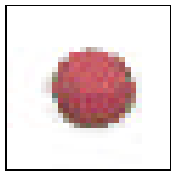


Abbildung 35

Bei niedrig gewählter Toleranz werden nicht alle Bildpunkte des Objektes erkannt. Ist z.B. ein Ball schlecht ausgeleuchtet (Abbildung 35), so hat der Schattenbereich eine dunklere Farbe und wird nicht erkannt (Abbildung 36).

Kann man die Toleranz nicht weiter erhöhen, da sonst zu viele Pixel außerhalb des Objektes erkannt werden, so ist eine zweite Suche erforderlich: Von dem in Abbildung 36 erkannten Punkthaufen läßt sich ein Mittelpunkt bestimmen, der uns eine ungefähre Position des Objektes liefert. In einem kleinen Bereich um diese Position soll erneut nach der Farbe gesucht werden. Die Toleranzeinstellung bei dieser zweiten Suche darf wesentlich höher liegen, da entfernte Bildbereiche mit ähnlichen Farben nicht mehr betrachtet werden (s. Prozedur Gezieltesuche2, Zeile325).

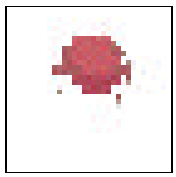


Abbildung 36

Wie groß der Durchmesser des Bereiches ist, in dem gesucht wird, hängt im wesentlichen von der Größe des Objektes ab und soll vom Anwender angegeben werden.

4.1.5 Mittelpunktsberechnung

Hat man die Bildpunkte eines annähernd kreisrunden Objektes gefunden, so werden von dem erkannten Punkthaufen nun die kleinsten und größten auftretenden X - und Y - Koordinaten gespeichert und anschließend der Mittelwert gebildet (Zeile 1053 - 1056, 1068 - 1069). Bei der Bildanalyse treten jedoch Ausreißer auf, d.h. es werden gelegentlich einzelne Pixel außerhalb des Objektes gefunden (siehe Xmax in Abbildung 37). Bei dieser einfachen Berechnung wirken sie sich stark auf die Position des Mittelpunktes aus.

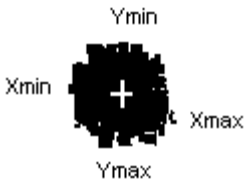


Abbildung 37

Daher soll bei der Zweitsuche ein anderes Verfahren dienen:

In jeder Zeile, in der Pixel gefunden wurden, wird ein eigener Zeilenmittelpunkt berechnet (durch rote Punkte in Abbildung 38 symbolisiert). Von allen Zeilenmittelpunkten kann anschließend ein Mittelwert gebildet werden (Zeile 385, 391). Somit erhält man eine X - Koordinate des Objektes, die durch einzelne Ausreißer nicht mehr stark verschoben werden kann.

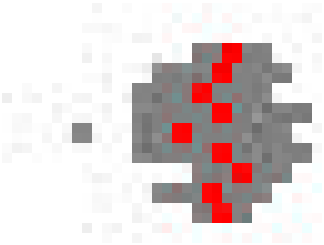


Abbildung 38

Zur Bestimmung der Y - Koordinate wird das gleiche Verfahren mit den Spalten durchgeführt.

4.1.6 Farbübernahme

Während der Aufnahme der Videofilme kann die Beleuchtung nie an allen Punkten der Bewegungsebene gleich stark sein. Dadurch verschiebt sich die Farbe des Objektes. Wird der Film nun durch eine Schleife vom ersten bis zum letzten Bild gesucht, so fallen nach einigen Bildern nur noch wenige Pixel in den Toleranzbereich der Suchfarbe (Abbildung 39 [21]).

Die Farbe, nach der gesucht wird, sollte von Bild zu Bild angepaßt werden.

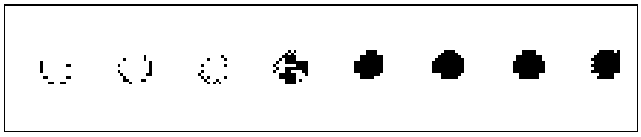


Abbildung 39

Nach gefundener Position des Objektes wird dazu aus dem Mittelpunkt eine neue Suchfarbe ausgelesen. Da sich hier aber zufällig eine völlig andere Farbe befinden kann - zum Beispiel durch eine Spiegelung auf dem Objekt -, sollte überprüft werden, ob sich die neue Farbe innerhalb des Farbtoleranzbereiches befindet (Prozedur Farbübernahme, Zeile 449 - 487). Bei gleicher Filmvorlage

[21] Die Bewegung erfolgte hier von rechts nach links

verbessert sich durch diese Prozedur das Ergebnis bedeutend (s. Abbildung 40).[22]

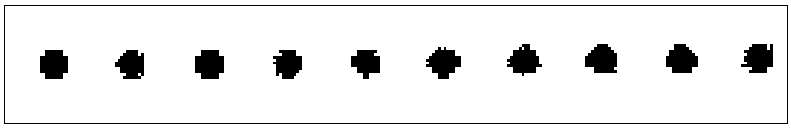


Abbildung 40

4.1.7 Suche beschleunigen

Bei hohen Auflösungen (z.B. 772x576 Pixel) benötigt die Analyse eines Bildes je nach Computer relativ viel Zeit. Bei jedem Bild werden dabei möglicherweise Bereiche untersucht, in denen sich das Objekt niemals aufhalten wird. Der Benutzer soll die Möglichkeit erhalten, solche Bereiche von der Analyse auszuklammern, indem er sie auf dem Videobild kennzeichnet. Das Programm muß dazu erst in einen anderen Modus gebracht werden (Zeilen 310 - 324), damit das Anklicken des Videos keine Farbe ausliest, sondern ein Rechteck zur Anzeige des Suchbereichs öffnet (Zeilen 1640 - 1624). Beim Verschieben der Maus wird das Rechteck dynamisch an die jeweilige Mausposition angepaßt (Zeilen 1660 - 1667). Beim Loslassen der Maustaste werden die Eckkoordinaten des Suchbereiches übergeben (Zeilen 1668 - 1705).
Damit können auch Bildbereiche, die eine ähnliche Farbe wie die des Objektes beinhalten, bewußt von der Analyse ausgeklammert werden.
Bewegt sich das Objekt aber über ein relativ großes x - y - Gebiet (z.B. bei einer Kreisbewegung), so kann der Bereich nur unwesentlich eingeschränkt werden. Für diesen Fall wurde eine zweite Möglichkeit geschaffen, den Suchbereich zu beschränken:

[22] Daß die gefundenen Pixel keine Kreise ergeben liegt an dem hohen Kompressionsgrad

Bei gleichförmig geradlinigen Bewegungen kann man die Position für das Bild Nr.i aber aus den Positionen der Vorbilder errechnen. Dazu muß man den Geschwindigkeitsvektor bestimmen:

$$\vec{x}_i = \vec{x}_{i-1} + \vec{v}t$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} v_x t \\ v_y t \end{pmatrix}$$

Bei gleichen Zeitabständen zwischen den Bildern ergibt sich:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} x_{i-1} - x_{i-2} \\ y_{i-1} - y_{i-2} \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 2x_{i-1} - x_{i-2} \\ 2y_{i-1} - y_{i-2} \end{pmatrix}$$

(s. Zeile 1225,1226)

Erfährt das Objekt jedoch eine Beschleunigung, so weicht es von dieser erwarteten Position ab. Wie groß der Radius der möglichen Abweichung ist, soll vom Benutzer angegeben werden.

Um die vorherberechnete Position wird (in den Zeilen 1227 - 1234) das Suchgebiet bestimmt. In Zeile 1238 werden nur Punkte innerhalb des Kreisgebietes zugelassen.

4.1.8 Berechnung von Geschwindigkeiten und Beschleunigungen

Um die gefundenen Koordinaten nicht in Pixel-, sondern in SI - Einheiten ausgeben zu können, wird in den Zeilen 1423 - 1491 mit Hilfe des Benutzers ein Eichfaktor errechnet. Damit können pro Bild die Zeit und die x - und y - Koordinate ausgegeben werden.

t_1	x_1	y_1
t_2	x_2	y_2
...	...	...
t_n	x_n	y_n

Aus diesen Werten erhält man durch Bildung des Differenzenquotienten die Geschwindigkeiten und Beschleunigungen.

Geschwindigkeit in x - Richtung: $v(x)_i = \frac{x_{i+1} - x_i}{\Delta t}$

Geschwindigkeit in y - Richtung: $v(y)_i = \frac{y_{i+1} - y_i}{\Delta t}$

Die Gesamtgeschwindigkeit läßt sich nun aus diesen beiden Werten berechnen:

$$v_i = \sqrt{v(x)_i{}^2 + v(y)_i{}^2}$$

(s Zeile 2032 - 2052)

Beschleunigung in x - Richtung: $a(x)_i = \frac{v(x)_{i+1} - v(x)_i}{\Delta t}$

Beschleunigung in y - Richtung: $a(y)_i = \frac{v(y)_{i+1} - v(y)_i}{\Delta t}$

Gesamtbeschleunigung: $a_i = \sqrt{a(x)_i{}^2 + a(y)_i{}^2}$

(Zeile 2053 - 2076)

Die Geschwindigkeit wurde damit zum Zeitpunkt $t_{v_i} = \frac{t_{i+1} - t_i}{2}$ bestimmt, die Beschleunigung zum Zeitpunkt $t_{a_i} = \frac{t_{vi+1} - t_{vi}}{2} = t_{i+1}$.

Die Daten werden in einer Tabelle gespeichert und angezeigt (Procedure Tabellefüllen, Zeile 2006).

4.1.9 Diagrammanzeige

Um die gefundenen Werte graphisch auftragen zu können, gibt es die Möglichkeit, über das OLE - Konzept[23]ein anderes Programm (z.B. Exel) zu benutzen. Da man jedoch nicht davon ausgehen kann, daß dieses spezielle Programm auf jedem Rechner installiert ist, wurde eine einfache Diagrammanzeige mittels

[23]Tabellen oder Diagramme können dabei von einem Programm (OLE - Server) in anderen Programmen (OLE - Client) eingebunden werden.

Delphi realisiert. Dazu eignet sich die *Chart* - Komponente. Man kann sich dem Zitat „Die Möglichkeiten der Chart - Komponente sind phantastisch, die Bedienung ist unmöglich“ [13] jedoch nur anschließen. Dies liegt daran, daß dieses Tool nicht von Borland, sondern von FX- Software entwickelt wurde und die Einbindung in die Delphi - Programmierumgebung nur teilweise geglückt ist[24]. Manche Steueranweisungen muß man über umständliche Zahlencodes übergeben.

In den Zeilen 1842 - 1874 und 1886 - 2004 werden je nach gewähltem Diagramm die Daten aus der zuvor erzeugten Tabelle ausgelesen. Nach dem Öffnen eines Informationskanals (z.B. Zeile 1900) müssen sie einzeln an die Chart - Komponente übermittelt werden.

4.1.10 Datenexport

Bei einigen analysierten Videosequenzen wird eine bloße Anzeige der Diagramme nicht ausreichen. So kann beispielsweise eine Glättung der Meßwerte oder das Erzeugen einer Fit - Funktion gewünscht sein. Diese Bearbeitungsmöglichkeiten werden von einigen anderen Programmen[25] geboten. Bevor die Daten zu diesen Programmen übermittelt werden, müssen sie in einem besonderem Format angeordnet werden.

Zur Trennung zweier Werte wird von fast allen Programmen das Tab - Zeichen akzeptiert. Bei dem Komma - Zeichen gibt es da schon mehr Probleme. Der internationale Standard sieht einen Punkt vor, während in deutschsprachigen Programmen ein Komma erwünscht ist.

In den Zeilen 2086 - 2121 werden die Daten in das jeweils gewünschte Format umgewandelt und in die Zwischenablage kopiert. In den Zeilen 2116 - 2118 werden sie zusätzlich in einer Datei gespeichert. Nach dem Aufrufen eines externen Programmes können sie so entweder mit Bearbeiten→Einfügen oder mit Datei→Öffnen importiert werden.

---

[24] das Fehlen einer Delphi - typischen Hilfe erschwerte das Programmieren erheblich.

[25] z.B. Easyplot, Stanfort Graphics ect.

Analysiert man Filme mit mehreren hundert Einzelbildern, so kann es bei dem Zwischenspeichern aller Werte für Koordinaten, Geschwindigkeiten und Beschleunigungen vor dem Export zu einem Speicherüberlauf kommen. Aus diesem Grund werden neben der Zeit nur die x/y- Koordinaten übertragen[26].

4.1.11 Stroboskopbild

Damit die abstrakten Datentabellen mit der anschaulichen Bewegung des Objektes im Videofilm leichter in Verbindung gebracht werden können, wurde die Darstellung des gesamten Bewegungsablaufes in einem Bild realisiert. Dazu wird zu Beginn ein komplettes Videobild in ein zweites Fenster kopiert (Zeile 546). Anschließend wird nach der Objektposition in dem Folgebild gesucht. Um den gefundenen Mittelpunkt des Objektes wird nun ein kleiner Bildbereich in das zweite Fenster kopiert (Zeile 620 - 641). Wird die Größe des kopierten Bereiches vom Benutzer richtig eingestellt, so erscheint nur das Objekt des zweiten Bildes in dem Vorbild. Nach Durchlaufen der Schleife (Zeile 552 - 677) zeigt sich das Objekt an allen Positionen der Folgebilder.

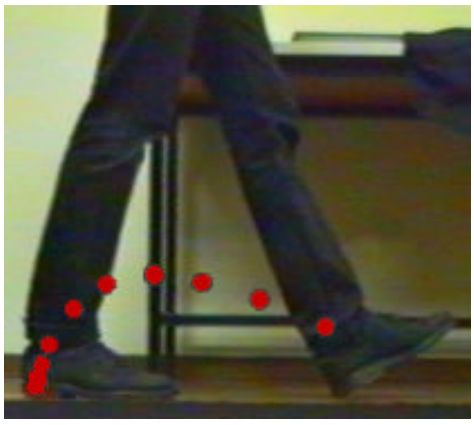


Abbildung 41 zeigt die Bewegung einer roten Markierung, die an meiner Schuhsohle angebracht wurde

Abbildung 41

[26]Sollen alle neun Werte (t,x,y,3·v,3·a) übertragen werden, muß in Zeile 2097 nur eine 9 anstatt einer 2 stehen.

4.1.12 Manuelle Positionserfassung

Wurden bei der Filmanalyse wesentliche Punkte des 3. Kapitels nicht beachtet, so unterscheidet sich die Markierung nicht ausreichend von ihrer Umgebung. Auch bei Filmen, die nicht selbst aufgenommen wurden (z.B. aus dem Internet) oder bei der Digitalisierung einer Fernsehsendung, schlägt eine automatische Analyse oftmals fehl. In solchen Fällen muß der Anwender die Filmposition selbst angeben.

Nachdem das Programm in den manuellen Modus gesetzt wurde (Zeile 869 - 881), bewegt er dazu den Mauszeiger an die richtige Position und klickt. Die Zeilen 2360 und 1492 - 1507 übergeben die Koordinaten des Mauszeigers in die Tabelle. (Was bei der automatischen Erfassung seitenweise Quelltext erforderte, steht hier durch die intelligente Mithilfe des Benutzers in ein paar Zeilen.)

Der Anwender hat die Möglichkeit, andere Mauszeigerformen zu wählen, um die Position optimal erfassen zu können. Der Mittelpunkt eines runden Objektes läßt sich mit einem runden Mauszeiger nun einmal genauer bestimmen. Dazu wurden selbstdefinierte Cursor in das Projekt mit eingebunden (Zeile 2321).

Das Videobild wird hierbei über den ganzen Bildschirm dargestellt (Zeile 872 - 875), um dem Benutzer bestmögliche Arbeitsbedingungen zu bieten.

Diese Großdarstellung ermöglicht außerdem Schülern, die weit entfernt vom Bildschirm sitzen, das Video noch zu erkennen.

4.2 Bedienung

In diesem Abschnitt soll die Oberfläche des Programmes und ihre Einstellmöglichkeiten dargestellt werden.

Bei der Entwicklung von **VIANA** wurde versucht, die Bedienung so einfach wie möglich zu halten. Vom Anwender wird lediglich vorausgesetzt, typische Windowselemente benutzen zu können.

Um die Übersichtlichkeit zu erhalten, werden während des Ablaufs alle nicht sinnvollen Einstellmöglichkeiten unsichtbar. Stellt man zum Beispiel die Option ‘Mittelpunkt anzeigen’ aus, so wird automatisch die Wahlmöglichkeit zwischen Kreuz oder Punkt ‘+ ·‘ unsichtbar.

Um den Umgang mit **VIANA** möglichst schnell zu lernen, wurden viele Buttons mit Symbolen versehen, welche die Funktion verdeutlichen sollen.

Bei kurzem Verweilen des Mauszeigers auf einem Element erscheint ein gelber Hilfetext zur Erläuterung.

Kompliziertere Eingaben (wie zum Beispiel das Kalibrieren des Bildschirmes) werden durch Hinweisfenster unterstützt, um den Bediener über den nächsten Schritt aufzuklären.

Das Wechseln des Mauszeigeraussehens soll den Benutzer über den momentanen Programmodus informieren. Auch bei der manuellen Positionserfassung ist der Wechsel zu einer anderen Zeigerform oftmals notwendig.

4.2.1 Film laden, Bedienung des Mediaplayers

Beim Starten des Programmes **VIANA** zeigt sich das Eröffnungsfenster mit der Möglichkeit, einen Videofilm zu laden. Durch einen Klick auf den Button öffnet sich das Standard - Windowsfenster zur Auswahl eines AVI - Filmes[27]. Bei der Wahl einer existierenden Datei wird der Film angezeigt. Sollte die Auflösung des Filmes größer als 288x384 Pixel sein, so wird er zur Anzeige auf diese Maße

Abbildung 42

[27]möglich sind alle von Windows lesbaren AVI - Filme, an einem Computer mit Digitalisierungskarte also auch das MJPEG - Format

verkleinert. Dürfte der Film einen größeren Platz am Bildschirm einnehmen, so würde er Bedienelemente verdecken. Zur späteren automatischen Analyse wird er aus Genauigkeitsgründen auf seine wahren Ausmaße gebracht.

Unter dem Video ist auch ein Bereich sichtbar, dessen Elemente den Umgang mit dem Video erlauben (Abb. 43).

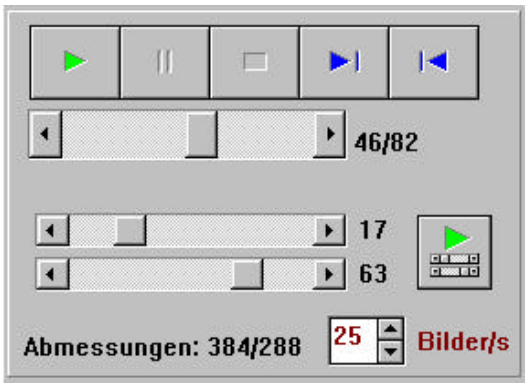


Abb. 43: Videosteuerung

In der obersten Reihe befinden sich die Standardbuttons *Play*, *Pause*, *Stop*, *Vorspulen bis Ende* und *Zurückspulen bis Anfang*. In der Laufleiste darunter wird die aktuelle Bildposition angezeigt. Mit dieser Leiste kann man per 'Drag and Drop' besonders bei längeren Filmen bequem Vor- oder Zurückspulen. Beim Klicken auf die Pfeile am Rand der Laufleiste wird die Filmposition um ein Bild bewegt.

Rechts daneben wird die aktuelle Bildnummer und die Gesamtanzahl der Einzelbilder des Filmes angezeigt.

Ein Videoclip zeigt eher selten vom ersten bis zum letzten Bild den interessanten Teil des physikalischen Ablaufs. Soll nur ein bestimmter Ausschnitt des Videos analysiert werden, so kann man mit den zwei schmaleren Laufleisten die Anfangs- und Endposition wählen. In Abb. 43 beginnt der zu analysierende Bereich bei Bildnummer 17 und endet bei Nummer 63. Bei der automatischen Analyse kann man den eingeschränkten Bereich jederzeit ändern.

Mit dem Playbutton kann man den eingeschränkten Filmbereich abspielen lassen. Im Gegensatz zum normalen Abspielen wird hierbei jedoch jedes einzelne Bild angezeigt. Dies kann unter Umständen einen Zeitlupeneffekt zur Folge haben.

Die zeitliche Auflösung des Filmes wird bei der Digitalisierung leider nicht gespeichert. Sie muß vom Benutzer angegeben werden, falls dieser bei der späteren Auswertung quantitative Ergebnisse wünscht. Da bei den meisten Filmen vermutlich der kleinstmögliche Zeitabstand zwischen Bildern vorliegen wird, ist 25 Bilder/s der Vorgabewert.

Es gibt nun zwei Möglichkeiten beim weiteren Vorgehen. Entweder erfaßt man die Position manuell (4.2.3.) mit Mausklick auf das Objekt oder man nutzt die Automatik (4.2.2.).

Abbildung 44

Über das Menü kann man auch zu einem späteren Zeitpunkt zu dieser Phase des Programmablaufes zurückzukehren.

4.2.2 Automatische Positionserfassung

Nach Betätigen des Buttons ‘Automatisch’ in Abbildung 44 werden neben einer Reihe von Buttons drei Bedienflächen angezeigt:

- Farbanzeige
- Sucheinstellungen
- Anzeige des Suchergebnisses.

4.2.2.1 Farbanzeige

Klickt der Anwender auf das Videofenster, so wird die Farbe des gewählten Pixels ausgelesen und in einem Farbblock angezeigt (in Abbildung 45 Grün). Nach dieser Farbe werden die Algorithmen das Video untersuchen. Die Anzeige der gewählten Farbe ist wichtig, da man in einem scheinbar einfarbigen Objekt schon einmal ein Pixel treffen kann, das sich deutlich von der gewünschten Farbe unterscheidet. Die Rot, Grün und Blauanteile der Farbe werden in einer Balkendarstellung mit einer Skala von 0 - 255 dargestellt. Die

Absolutwerte der Farbanteile sind in den Editierfeldern daneben zu sehen. Bei Bedarf kann der Anwender die Suchfarbe hier von Hand ändern.

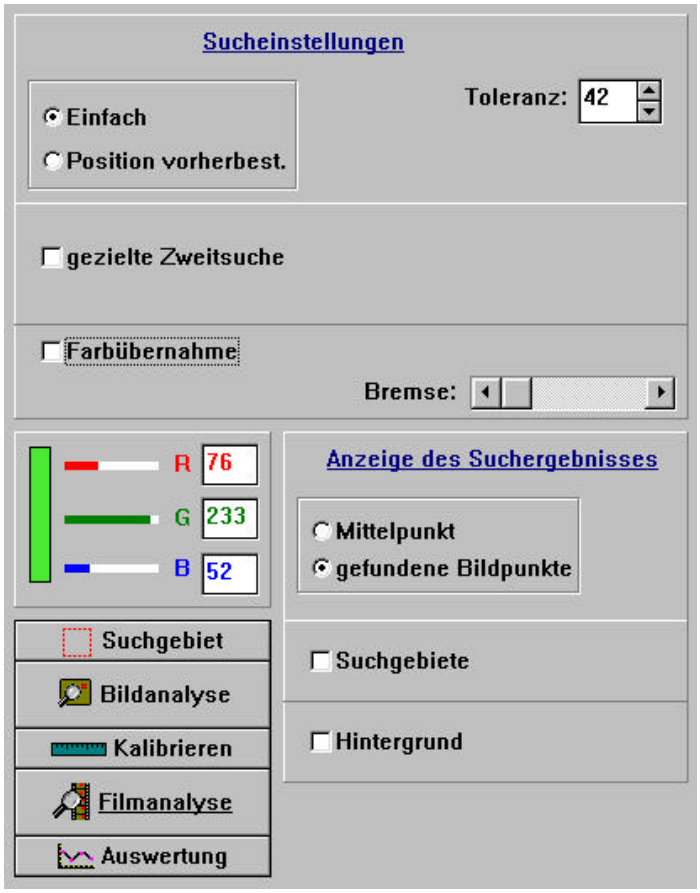


Abbildung 45

4.2.2.2 Anzeige des Suchergebnisses

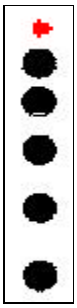


Abb. 46

Hat das Programm nach dem Starten der Analyse Bildpunkte gefunden, so kann man diese mit der Option '*gefundene Bildpunkte*' vollständig anzeigen lassen. Dazu öffnet sich ein weiteres Fenster mit den Abmessungen des Videos. Rote Punkte zeigen erkannte Pixel bei der Erstsuche. Ist die Option '*ÿ gezielte Zweitsuche*' aktiviert, so werden diese von schwarzen Bildpunkten übermalt. Durch das Anzeigen aller erkannten Pixel kann man während der Analyse kontrollieren, ob die Toleranzeinstellungen richtig getroffen wurden und die Abbildung der Markierung annähernd rund ist.

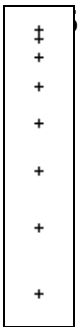
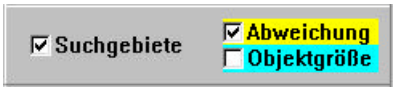


Abb. 48

Alternativ dazu kann man sich den erkannten Mittelpunkt anzeigen lassen. Dies ist dann sinnvoll, wenn sich die Position der Objekte nur wenig ändert. Bei der Mittelpunktanzeige kann man wiederum zwischen Kreuzen (Abb. 48) oder Einzelpunkten wählen (vgl. Abbildung 70).

Um zu überprüfen, ob die Sucheinstellungen richtig getroffen wurden, aktiviere man die Option '*ÿ Suchgebiete*'. In dem Ergebnisfenster werden die Gebiete farbig gekennzeichnet, in denen nach Bildpunkten gesucht wird (siehe Abbildung 47).



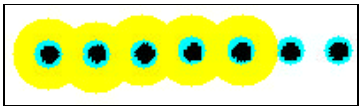


Abbildung 47

Möchte der Anwender die Umgebung der Suchanzeige nicht weiß lassen, so kann er mit '*ÿ Hintergrund*' ein Originalbild auf das Suchergebnisfenster kopieren lassen. Die Darstellung verliert so an Abstraktheit (siehe auch 'Stroboskopbild').

Möchte man bei schnellen Computern die Anzeige der Suchergebnisse genauer verfolgen, so kann man durch 'ziehen der Bremse' die Darstellungszeit verlängern. Es handelt sich dabei nicht

um eine sinnlose Warteschleife, sondern es wird Windows mehr Zeit zum Bearbeiten von Systemfunktionen überlassen. Man merkt es daran, daß das Programm während der Filmanalyse nicht so sehr in seine Rechnungen vertieft ist, sondern nun besser auf Eingaben des Anwenders reagiert. Dieser hat die Möglichkeit, auch nach dem Start der Analyse noch Werte zu ändern[28]. Hat das Programm mit einem Bild des Filmes besondere Erkennungsprobleme, so kann man ein Bild zuvor noch dynamisch Werte ändern.

4.2.2.3 Sucheinstellungen

Um den Umgang mit der automatischen Analyse zu erlernen, fängt man am Besten mit der durchschaubarsten Einstellung an. Dazu aktiviert man die

---

[28]daher auch die Form des Mauszeigers, eine Mischung zwischen normalem Pfeil und der Sanduhr, die 'Beschäftigtsein' signalisiert

Option ‘ðEinfach’ und deaktiviert ‘*ðgezielte Zweitsuche*’ und ‘*ðFarbübernahme*’ (siehe Abbildung 45, oben). Der einzig einstellbare Wert ist jetzt die Toleranz.

Nachdem die Objektfarbe durch Anklicken des Videos gewählt wurde, kann man das aktuelle Einzelbild mit der Toleranz durchsuchen lassen. Dazu dient

der Button ‘Bildanalyse’. Nach Betätigung öffnet sich das Suchergebnisfenster. Man schiebt das Fenster am besten an eine Position auf dem Bildschirm, wo es am wenigsten stört.

Angezeigt werden die Pixel in der Farbe der Filmvorlage.

Ist die Toleranz zu niedrig eingestellt, so zeigen sich nur einige Pixel des Objektes (Abb.51). In dem Falle erhöhe man die Toleranz und klicke erneut ‘Bildanalyse’. Wurde der Wert zu hoch gewählt, so zeigen sich außerhalb der Markierung noch erkannte Pixel (Abb.49). Man muß die Toleranz also wieder verkleinern, bis nur Bildpunkte innerhalb des Objektes gefunden wurden (Abb.50). Von dieser Einstellung sollte man jedoch etwas nach unten abweichen, um eine problemlose Erkennung der Objektposition in den Folgebildern zu ermöglichen.

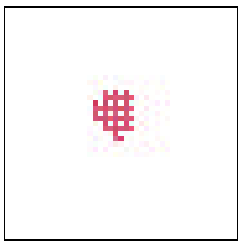


Abb.51

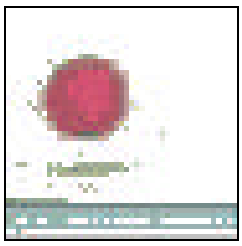


Abb.49

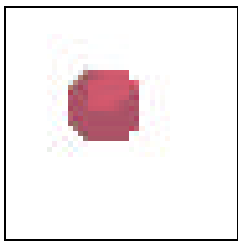


Abb.50

Startet man nun die Analyse des Filmes, so könnte sich je nach Film ein Suchergebnis wie in Abb. 52 zeigen. Das erste Bild ist in diesem Falle unten. Ab dem vierten Bild trat durch schlechte Beleuchtungsverhältnisse offenbar eine Farbveränderung ein. Nach dem Einschalten der '*ðFarbübernahme*' ist das Ergebnis schon besser (Abb. 53). Eine aktivierte Farbübernahme ist bei fast allen Filmen sinnvoll. Sollte die Farbe, die aus der Markierung übernommen werden sollte, zu sehr von der Suchfarbe des Vorbildes abweichen, so gibt das Programm einen Signalton aus und die alte Suchfarbe bleibt aktuell.[29]

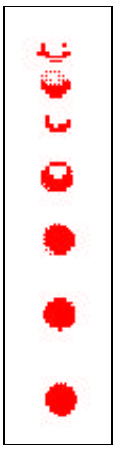


Abb. 52

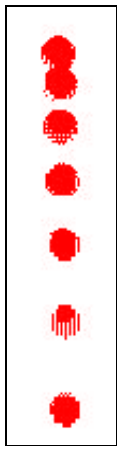


Abb. 53

Eine genaue Bestimmung des Mittelpunktes ist mit diesen Einstellungen aber noch nicht zu erreichen, da die Markierung oft nicht vollständig erkannt wird (Abb. 53). Da eine Erhöhung der Toleranz nicht möglich ist, aktiviere man nun die Option *'ðgezielte Zweitsuche'*. Es zeigen sich zwei Eingabefelder,

die Toleranz für die Zweitsuche und die Objektgröße. Durch die bisherige Erstsuche wurde ein ungefährer Mittelpunkt des Objektes gefunden. Um diesen Punkt soll in einem kleinen Gebiet noch einmal genauer nach der Farbe gesucht werden. Wie groß dieses Gebiet ist, richtet sich nach den Ausmaßen der Markierung und wird im unteren Feld eingestellt. Mit welcher Toleranz die Zweitsuche stattfinden soll, kann man im oberen Feld eintragen.

Vorgehensweise zur richtigen Wahl der Parameter:

Zuerst ermittele man die Toleranz, bei der alle Pixel der Markierung erkannt werden. Es spielt dabei keine Rolle, ob in einiger Entfernung zum Objekt fremde Bildpunkte erkannt werden. Diese Gebiete werden von der Zweitsuche

---

[29] Dies kann der Fall sein, wenn in Mittelpunksnähe einige Pixel eine falsche Farbe haben, zum Beispiel bei einer Spiegelung der Lampe auf einem glänzendem Ball.

nicht betrachtet. In unmittelbarer Nähe des Objektes dürfen dagegen keine Fehler auftauchen.

Den so ermittelten Wert trage man bei ‘Zweittoleranz’ ein. Der Wert der Ersttoleranz sollte nun deutlich darunterliegen, er wird nur benötigt, um (im Extremfall) mindestens einen Pixel des Objektes zu finden.

Zur Kontrolle, ob die Größe des Suchgebietes richtig gewählt wurde, aktiviere man unter *Suchgebiete®Objektgröße* die Anzeige des türkis dargestellten Bereiches. In Abb. 54 wurde die Filmanalyse beim fünften Bild unterbrochen, bevor die Zweitanalyse das rote Suchergebnis der Erstanalyse übermalen konnte. (In dem Beispiel waren die Einstellungen: Toleranz: 20, Zweittoleranz: 100, Objektgröße: 32 .)

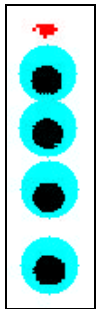


Abb. 54

Wählt man anstelle der einfachen Suche die Option ‘*Position vorherbestimmen*’, so zeigt sich das letzte Eingabefeld[30].

Mit dieser Option hat man die Möglichkeit, den Analyseprozeß zu beschleunigen. Durch Errechnen eines Geschwindigkeitsvektors kann die erwartete Position vorhergesagt werden. Durch das Eingabefeld hat man die Möglichkeit, die Größe des Suchgebietes um diesen Punkt einzustellen. Je nachdem, wie groß die Beschleunigung auf das Objekt ist, weicht es von dieser Position ab.

Um die Einstellung zu kontrollieren, aktiviere man unter *Suchgebiete ® Abweichung* die Anzeige des gelben Gebietes. Die ersten beiden Bilder können nicht vorhergesagt, daher auch keine Gebiete gezeichnet werden. In Abbildung 55 wurde der Wert der möglichen Abweichung aus Darstellungsgründen sehr klein gewählt. Er sollte aber eher zu groß als zu klein eingestellt sein, damit sich das Objekt nicht außerhalb des Suchgebietes befindet. Besonders bei

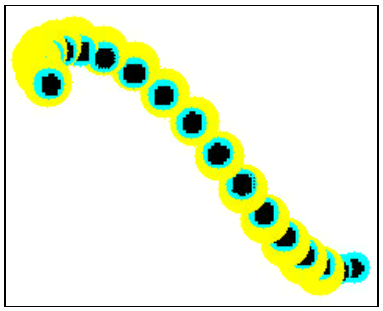


Abbildung 55

Stoßvorgängen kann man die auftretenden Beschleunigungen leicht unterschätzen.

4.2.2.4 Suchgebiet einschränken

Erfolgt die Bewegung des Objektes nur über einen beschränkten Bereich des Videofensters, so ist es sinnvoll, nach Drücken des Buttons ‘*Suchgebiet*’ diesen zu kennzeichnen. Die Suche beschränkt sich dann nur auf dieses Gebiet. Neben den Geschwindigkeitsvorteilen kann man so auch Bildbereiche, die kritische Farben beinhalten (der Markierungsfarbe zu ähnlich) unberücksichtigt lassen. Zum Kennzeichnen zieht man mit gedrückter Maustaste ein Rechteck auf dem Video.

4.2.2.5 Kalibrieren

Um bei der späteren Auswertung auch quantitative Aussagen machen zu können, muß das Programm einen Eichfaktor erhalten. Mit diesem kann es dann von Pixeleinheiten in reale Längenangaben umrechnen. Wird diese Kalibrierung nicht durchgeführt, so ist der Eichfaktor eins, was bedeutet, daß ein Pixel einem Meter entspricht. Für rein qualitative Aussagen ist dies unbedeutend. Nach dem Anklicken des Buttons ‘*Kalibrieren*’ ändert sich die Mauszeigerform und der Benutzer wird aufgefordert, die beiden Endpunkte einer Strecke bekannter Länge anzuklicken. Dann wird eine Angabe über die Länge gefordert. Zum Schluß wird gefragt, ob der Nullpunkt verschoben werden soll. Falls nicht, befindet er sich in der linken unteren Ecke. Die Koordinaten der Objektposition werden mit diesen Angaben berechnet. Diese gefundenen Koordinaten kann

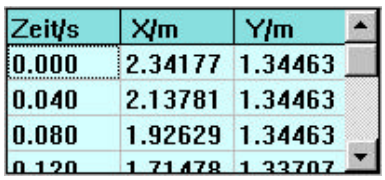


Abbildung 56: Meßwerttabelle

[30] Alle bisherigen Eingabefelder behalten ihre Bedeutung.

man in der Tabelle betrachten. Der nächste Schritt wäre nun die Auswertung (Kap. 4.2.4.).

4.2.3 Manuelle Positionserfassung

Bei Filmen, die nicht zu einer automatischen Analyse geeignet sind, kann man die Position des Objektes auch per Mausklick angeben. Auch Filme, die nur aus sehr wenigen Einzelbildern bestehen, können durch diese Möglichkeit vielleicht schneller analysiert werden.

Nach dem Laden des Filmes sollte man zuerst das Start- und Endbild festlegen (s.o.). Betätigt man den Button ‘*Manuell*’, so öffnet sich ein neues Fenster und das Videobild wird unabhängig von seiner Originalgröße über den ganzen Bildschirm dargestellt. Besitzt das Video nur eine geringe Auflösung von beispielsweise 160x120 Pixel, so erscheint es hier sehr grobkörnig. Um das Video ein Bild weiterzuspulen, drücke man die rechte Maustaste. Das Vorbild erreicht man mit dem Button ‘*-1*’. Bevor man mit der Positionserfassung beginnt, sollte man auch hier erst den Bildschirm kalibrieren (s. 4.2.2.5.). Nachdem der Eichfaktor bestimmt wurde, kann man mit der linken Maustaste den Mittelpunkt des Objektes an- und danach ein Bild weiterklicken. Der Vorgang kann jederzeit abgebrochen werden, es darf aber kein Bild ausgelassen werden. Der Anwender wird sonst darauf hingewiesen, daß bei der folgenden Auswertung undefinierte Werte auftauchen werden.

Abbildung 57

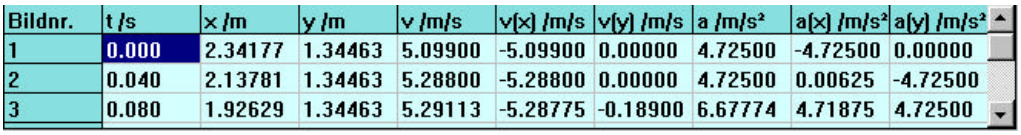
4.2.4 Auswertung

Wenn man bei der automatischen oder der manuellen Analyse den Button ‘*Auswertung*’

anklickt, so öffnet sich ein Fenster zur Anzeige von Diagrammen. Voraussetzung ist, daß zuvor mindestens drei Positionen erfaßt wurden.

Im unteren Teil des Fensters befindet sich die Tabelle mit den Eintragungen der Zeitwerte, der gefundenen Koordinaten, der berechneten Geschwindigkeiten und Beschleunigungen.

Bei den Spalten für die Geschwindigkeiten existiert natürlich ein Wert weniger, bei den Beschleunigungsspalten zwei Werte weniger als in den ersten



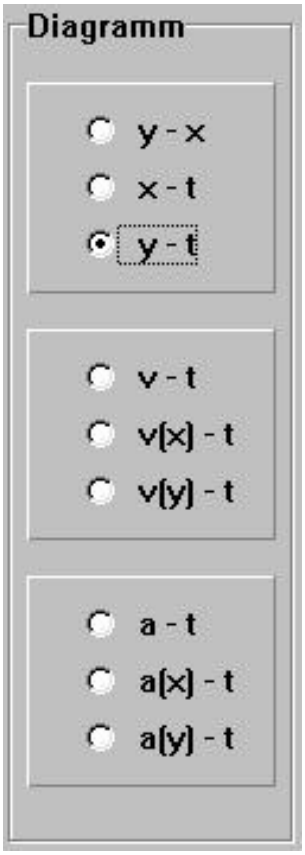
Bildnr.	t /s	x /m	y /m	v /m/s	v[x] /m/s	v[y] /m/s	a /m/s²	a[x] /m/s²	a[y] /m/s²
1	0.000	2.34177	1.34463	5.09900	-5.09900	0.00000	4.72500	-4.72500	0.00000
2	0.040	2.13781	1.34463	5.28800	-5.28800	0.00000	4.72500	0.00625	-4.72500
3	0.080	1.92629	1.34463	5.29113	-5.28775	-0.18900	6.67774	4.71875	4.72500

Abb. 58: Koordinaten, Geschwindigkeiten und Beschleunigungen

Spalten.

4.2.4.1 Diagramme

Auf der linken Seite des Fensters kann man das darzustellende Diagramm wählen. Neun Möglichkeiten stehen zur Wahl:



- Ortsdiagramm
- Änderung des X- Wertes über der Zeit
- Änderung des Y- Wertes über der Zeit

- Gesamtgeschwindigkeit über der Zeit
- Geschwindigkeit in x- Richtung über der Zeit
- Geschwindigkeit in y- Richtung über der Zeit

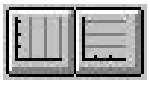
- Gesamtbeschleunigung über der Zeit
- Beschleunigung in x- Richtung über der Zeit
- Beschleunigung in y- Richtung über der Zeit

Abbildung 59

4.2.4.2 Diagrammformat

Das zuerst angezeigte Diagramm ist die y-x -Darstellung der Meßwerte. Über dem Diagramm befinden sich einige Werkzeugbuttons. Wählt man eine andere Darstellung als die Ortskurve, so erweitert sich die Buttonleiste um weitere fünf. Alle Buttons dienen zum Ändern des Diagrammformates:

Abbildung 60



Mit diesen Buttons kann ein horizontales oder vertikales Raster eingeblendet werden (siehe Abbildung 64).

Hierbei öffnet sich ein Standardfenster zum Einstellen verschiedener Optionen:

- Skalieren der Achsen
- Aussehen der Markierungspunkte (Kreis, Dreieck, Quadrat)
- Anzeigen der Nullinie (s.Abbildung 63, Abbildung 66)
- Anzeige von Werten neben den Markierungspunkten (s. Abbildung 65)

Das Diagramm erhält eine räumliche Tiefe

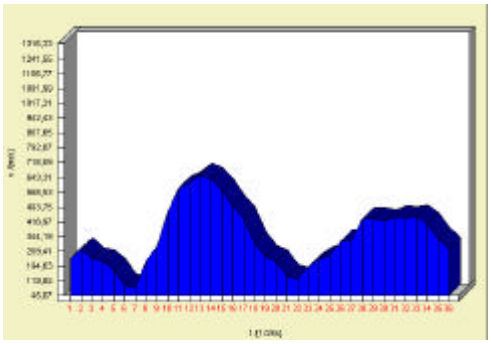


Abbildung 61

Flächendiagramm: Die Fläche zwischen Kurve und Nullinie wird gefüllt.

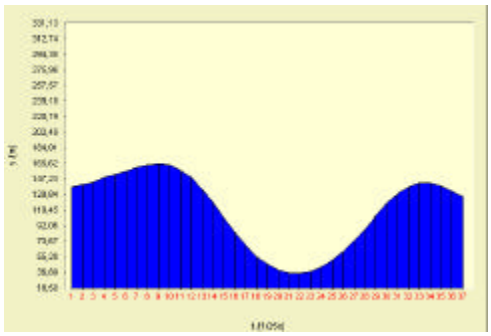


Abbildung 62

Balkendiagramm

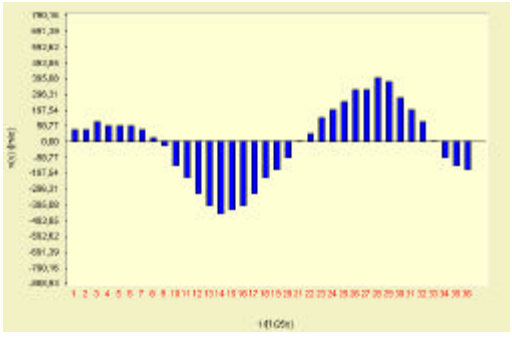


Abbildung 63

Liniendiagramm

(Hier mit aktiviertem horizontalem Raster)

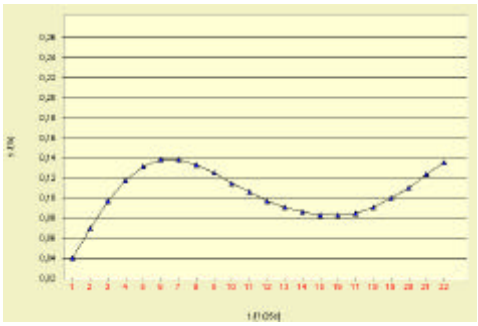


Abbildung 64

Punktdiagramm

(Hier mit aktivierter Anzeige der Absolutwerte)

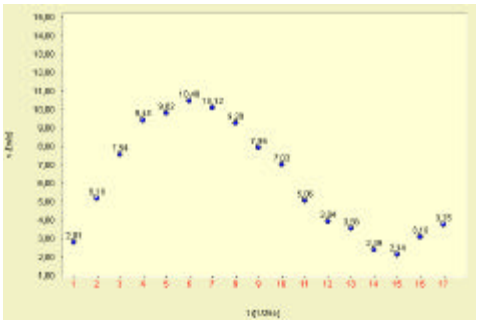


Abbildung 65

Splinediagramm

Die gezeichnete Kurve gleicht große Sprünge etwas aus.

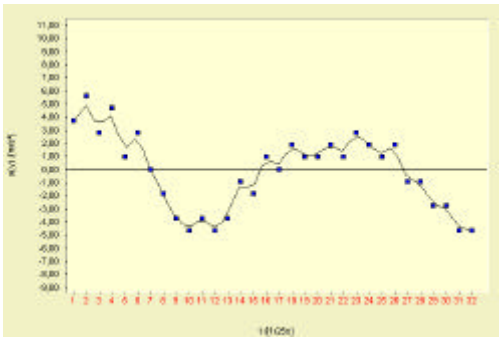


Abbildung 66

4.2.4.3 Datenexport

Durch Drücken des Buttons ‘Speichern’ wird die Tabelle in die Zwischenablage kopiert. Der Anwender wird anschließend gefragt, ob noch eine Datei erstellt werden soll. Beim Bestätigen kann er über ein Stan-

dardfenster Name und Pfad angeben. Ist die Option ‘*Exelformat*’ aktiv, so werden die Nachkommastellen durch ein ‘,’ getrennt, ansonsten durch ein ‘.’.

4.2.5 Stroboskopbild

Um eine Grafik zu erhalten, die den ganzen Bewegungsablauf in einem Videobild darstellt, kann man im Menü→Extras ‘Stroboskopbild’ anklicken. Damit zeigt sich am unteren Bildschirmrand ein neues Bedienelement (Abb. 67).

Abb. 67

Nach Betätigen des Buttons wird zuerst das aktuelle Bild als Hintergrund auf das Suchergebnisfenster kopiert. Bei richtiger Einstellung der Suchparameter wird aus den Folgebildern nur das Objekt auf diesen Hintergrund gezeichnet. Sollen dabei nicht die Originalfarben aus den Videobildern verwendet werden, so aktiviere man ‘*Objekt einfarbig*’. Hier wird entweder Rot oder bei aktivierter Option ‘*Mittelpunktfarbe*’ die Suchfarbe des aktuellen Bildes verwendet.

Die Option ‘mit Rand’ bewirkt, daß um die gefundenen Pixel noch ein weißer Kreis gezeichnet wird (siehe Abbildung 69). Dies empfiehlt sich immer dann, wenn sich die Position des Objektes nur wenig ändert und durch das Überlagern der Folgebilder die Objektgrenze nicht mehr zu erkennen ist.

In Abbildung 68 wurde durch geeignete Wahl des Parameters ‘Objektgröße’ das Randgebiet des pinkfarbenen Flummis ein wenig mitgezeichnet. Sie verlaufen so in den Umkehrpunkten nicht zu einem Streifen (auch ohne das Zeichnen des weißen Randes). Daß der Flummi hier einfarbig aussieht, liegt an der Zusatzbeleuchtung von links, die einen Schattenwurf verhindern soll.

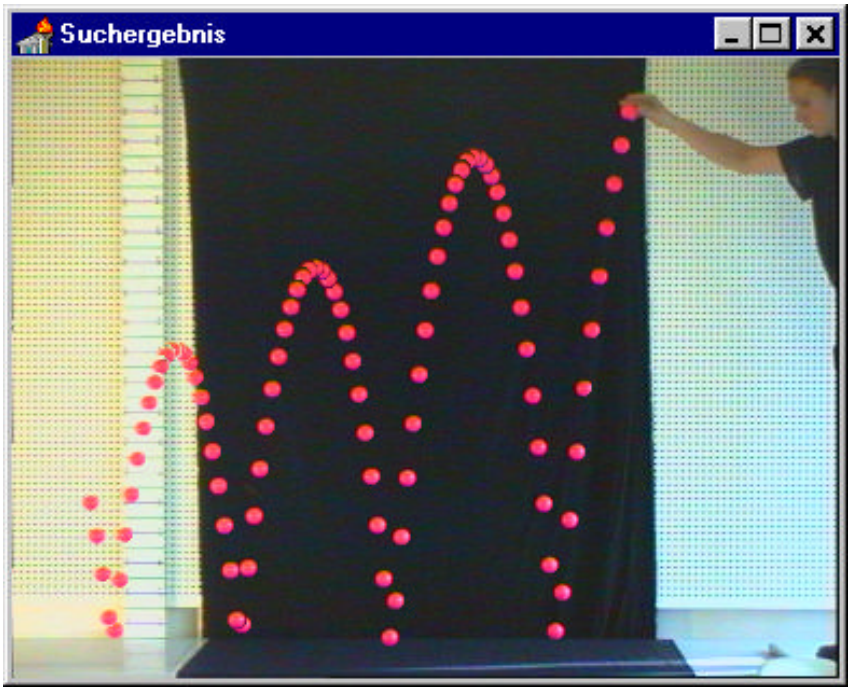


Abbildung 68

5 Versuchsdurchführungen

In diesem Kapitel soll das entwickelte Meßwert - Erfassungssystem durch das Analysieren und Auswerten von einigen exemplarischen Bewegungsabläufen erprobt werden.

Um Aussagen über den Fehler bei der Positionserfassung zu erhalten, wurde zu Beginn eine rotierende Markierung aufgenommen. Die Mittelpunkte der Markierungsfläche müssen sich auf einem Kreis befinden. Durch die Abweichung der erkannten Mittelpunkte von dem Kreis kann der Fehler bestimmt werden.

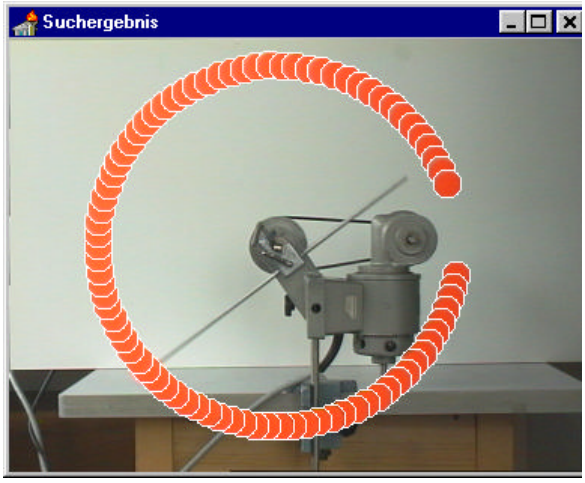


Abbildung 69

Die Bewegung wurde mit 1/1000s Belichtungszeit aufgenommen, die Helligkeit (Exposure) um zwei Stufen erhöht, Focus und Weißabgleich - Automatik deaktiviert und die Sättigung mit der Digitalisierungssoftware auf den Wert 220[31] eingestellt.

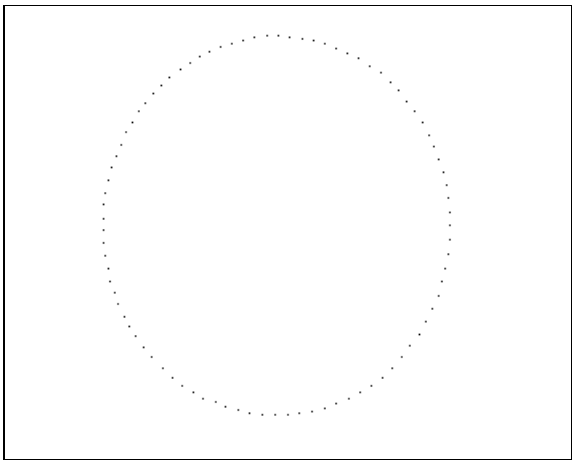


Abbildung 70

In dem Analyseprogramm **Viana** wurde als Anzeige des Suchergebnisses die Mittelpunktsdarstellung als Punkt gewählt.

Abbildung 70 zeigt das Ergebnis von ca. 80 erkannten Mittelpunkten bei einmaliger Umdrehung.

[31] maximal möglich ist 255

Läßt man die Markierung sehr lange rotieren, so sollte durch die Überlagerung ein geschlossener Kreis entstehen. Tatsächlich wurde jedoch nur ein Film mit 750 Einzelbildern aufgenommen und analysiert (Abbildung 71). Die entstandene Filmdatei hatte eine Größe von 49 Megabyte.

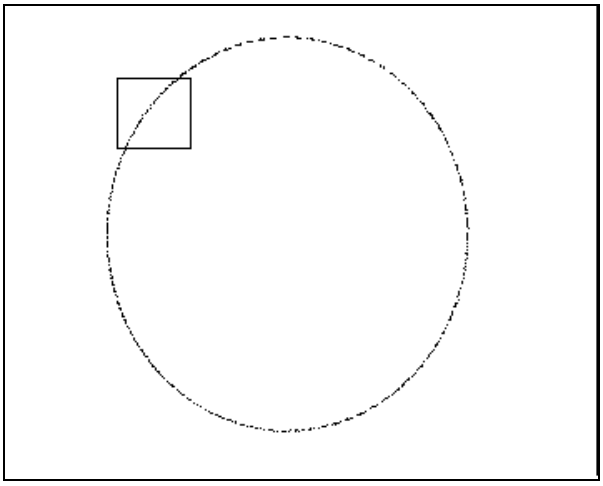


Abbildung 71

Abbildung 72 zeigt einen vergrößerten Ausschnitt des Suchergebnisses. Außerdem wurde nachträglich ein roter Kreis eingeblendet. Er hat ungefähr den Rotationsradius und soll die Abweichung von erkannten zu tatsächlichen Mittelpunkten verdeutlichen. In den meisten Fällen trifft das Meßsystem (unter Berücksichtigung der Quantisierung) die korrekten Koordinaten. In einigen Fällen liegen sie jedoch ein Pixel daneben.

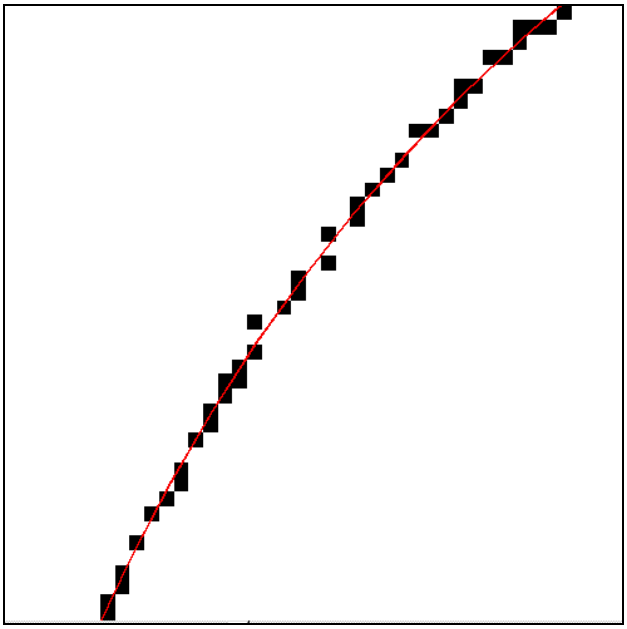


Abbildung 72

Diesen Fehler bei der Erfassung der Meßwerte muß man je nach Qualität des vorliegenden Filmes neu bewerten. Zum Beispiel kann bei einem schlecht ausgeleuchteten Ball mit Schattenwurf der erkannte Mittelpunkt auch drei Pixel oder mehr abweichen.

Bei nur einer Lichtquelle kann der Schatten während der Bewegung wandern.

Auch solche systematischen Fehlerquellen müssen je nach Fall beurteilt werden.

Beim manuellen Erfassen der Position durch Mausklick muß auch ein Fehler von einem Bildpunkt angenommen werden.

Außerdem gibt es einen Fehler beim Kalibrieren des Bildschirmes. Beim Anklicken einer bekannten Strecke muß von einer Genauigkeit von ±1 Pixel ausgegangen werden.

Ein Fehler durch ungenaue Eichstrecken kann in der Regel vernachlässigt werden. Diese können zum Beispiel auftreten, wenn eine Meßlatte nicht parallel zur Kameraebene aufgestellt wird.

Auch Fehler durch Kissen- oder Tonnenverzerrungen der Kamera sind bei dem Vermeiden von Extremzooms im Verhältnis zu den genannten Fehlern unbedeutend.

5.1 Der schräge Wurf

Um eine Überprüfung der Meßmethode durchzuführen, eignet sich der freie Fall besonders. Zum einen ist das theoretisch erwartete Ergebnis genau bekannt, zum anderen steigen die Geschwindigkeiten der Bewegung von sehr kleinen zu relativ großen Werten an[32]. Da der freie Fall im Labor jedoch ein relativ kurzer Bewegungsablauf ist, erhält man auch nur wenige Meßwerte (maximal 25 Werte/s). Bei einem senkrechtem oder schrägem Wurf dagegen, läßt sich die Zeit, in der das Objekt der Erdanziehung ausgesetzt ist, bei gleicher Fallhöhe verdoppeln. Aus diesem Grund wurde ein springender Flummi während der Auf- und Abwärtsbewegung mit 25 Bildern/s gefilmt

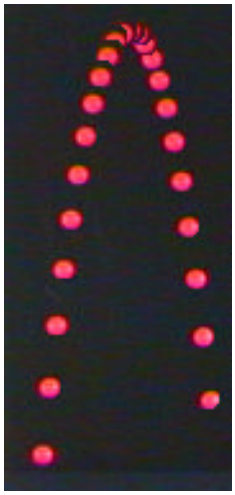


Abb.73: M-JPEG

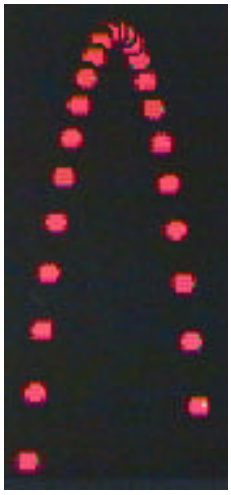


Abb.74: Cinepak

[32]auch hier ist wieder die Geschwindigkeit des Objektes in Relation zur Bildschirmhöhe gemeint.

und analysiert. Die Auflösung beträgt 384x288 Pixel. Der Film besteht aus 26 Einzelbildern. Die letzten zwei Bilder zeigen den Flummi nach dem Bodenkontakt.
Zur Kalibrierung wurde parallel zur Kameraebene eine Meßlatte aufgestellt. Der Hintergrund ist mit schwarzem Stoff abgedunkelt, um einen möglichst großen Kontrast zum Objekt zu gewährleisten. Der Film Flummi.avi befindet sich auch auf der beiliegenden Diskette.

Er wurde mit dem gängigen Softwarecodec Cinepak komprimiert (s. Kap. 2.1.2.5.). Der Originalfilm (M-JPEG) hat eine Größe von 2 MB, der Cinepakfilm dagegen nur 0,85 MB. Durch die Kompression ergeben sich einige Informationsverluste und somit Nachteile bei der Auswertung. Abb.73 zeigt das Stroboskopbild des Bewegungsablaufes. In Abb.74 lassen sich die Verluste erkennen. Der Flummi wird in einigen Bildern nicht mehr rund, sondern eckig dargestellt. Da der M-JPEG - Film aber nur auf dem Rechner mit der Digitalisierungskarte abspielbar ist, muß man die Ungenauigkeiten bei Cinepakfilmen auf anderen Computern hinnehmen.
Die Bewegungsanalyse soll im folgendem an dem Originalfilm durchgeführt werden. Der Leser kann sie durch den Film auf der Diskette nachvollziehen (siehe auch ausführliche Anleitung, Anhang B).
Die Luftreibung wird vernachlässigt, da die Fallhöhe nur 1,5 m betrug. Das Ortsdiagramm zeigt die gefundenen Koordinaten (Abbildung 75).

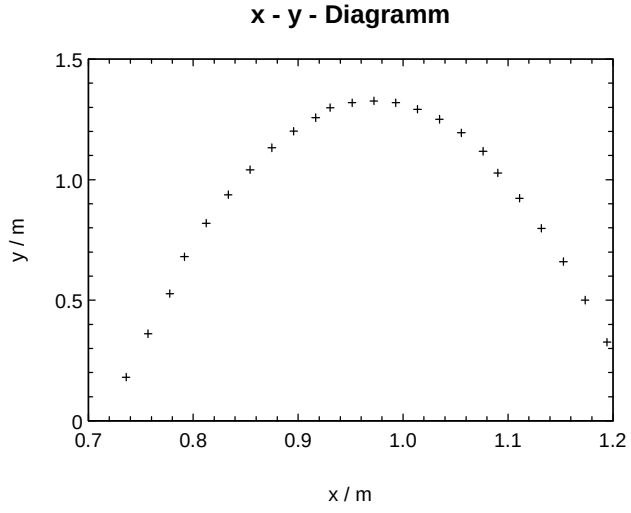


Abbildung 75

Die Bewegung läßt sich nach dem Superpositionsprinzip in eine x - und eine y - Bewegung aufteilen. Da nur Kräfte in y -

Richtung wirken, muß die x - t - Kurve eine Gerade zeigen:

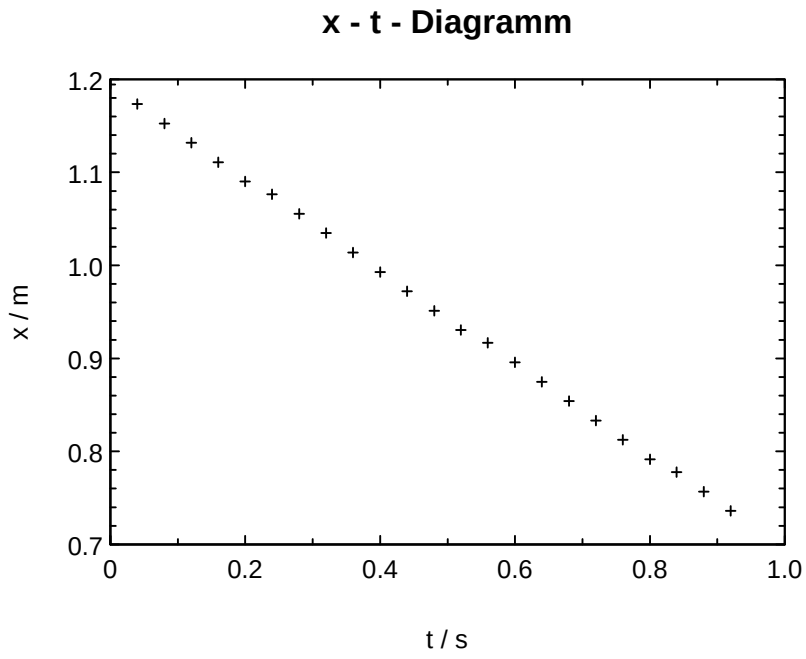


Abbildung 76

Nach dem Fallgesetz müssen die gefundenen Y- Werte auf einer quadratischen Funktion liegen.Die Meßwerte y(t) sind in Abbildung 77 dargestellt.[33]

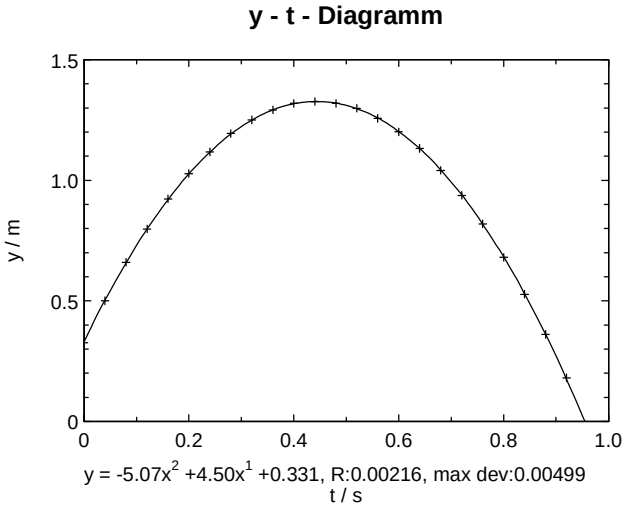


Abbildung 77

Durch das automatische Anpassen eines Polynoms zweiten Gerades[34] kann man die Abweichung der Meßpunkte von dem theoretischen Verlauf überprüfen.

Die Übereinstimmung der Meßwerte mit der Fit - Kurve

$y = -5{,}07 \cdot (t - 0{,}44)^2 + 1{,}33$

[33]nicht zu verwechseln mit dem Ortsdiagramm

[34]die Daten wurden über ‘Speichern’ exportiert und in dem Programm Easyplot untersucht. Dieses bietet die Möglichkeit, Fit - Kurven berechnen zu lassen.

(Gleichung 1)

kann vom Leser selbst beurteilt werden.

Läßt man das v(y)-t- Diagramm anhand der erfaßten Meßwerte darstellen, so kann man die erwartete Gerade ungefähr bestätigen[35].

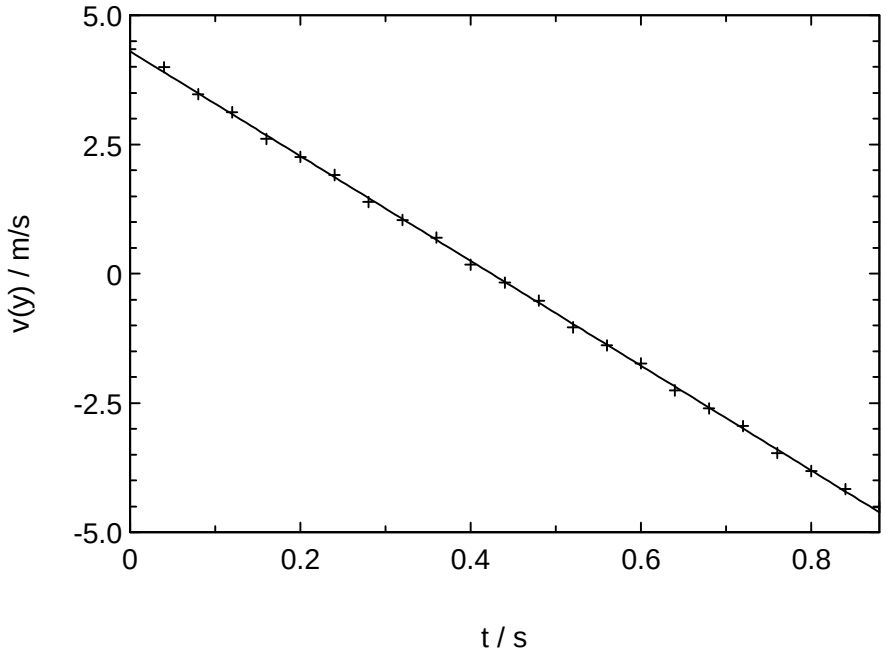


Abbildung 78: v(y) - t - Diagramm

Der Wert der Fallbeschleunigung läßt sich durch die Steigung der Geraden in Abbildung 78 oder durch zweimaliges Ableiten der Fit - Funktion (Gleichung1) bestimmen. In beiden Fällen ergibt sich g = 10,14 m/s² (3% Abweichung zum Literaturwert).

Die x,y - Koordinaten können nur die Position eines bestimmten Pixel haben. Zwischenwerte sind nicht möglich. Die kleinste bestimmbare Einheit (die Strecke zwischen zwei Pixeln) ist in diesem Fall:

$$x_{Einheit} = \frac{1{,}8m}{253Pixel} = 7{,}1\text{mm}$$

[35]Zur Berechnung der Geschwindigkeiten und Beschleunigungen siehe Kap 4.1.8.

In dem y-x- Diagramm sind derart kleinen Strecken kaum erkennbar. Bei den Geschwindigkeitsdiagrammen wirken sich die diskreten Werte jedoch stärker aus:

$v_{Einheit} = \frac{x_{Einheit}}{\Delta t} = 0{,}18$ m/s

In Abbildung 78 wird die Rasterung schon sichtbar.

Noch stärker wirkt sich dies auf die Bestimmung der Beschleunigungen aus:

$a_{Einheit} = \frac{v_{Einheit}}{\Delta t} = 4{,}4 \text{m/s}^2$

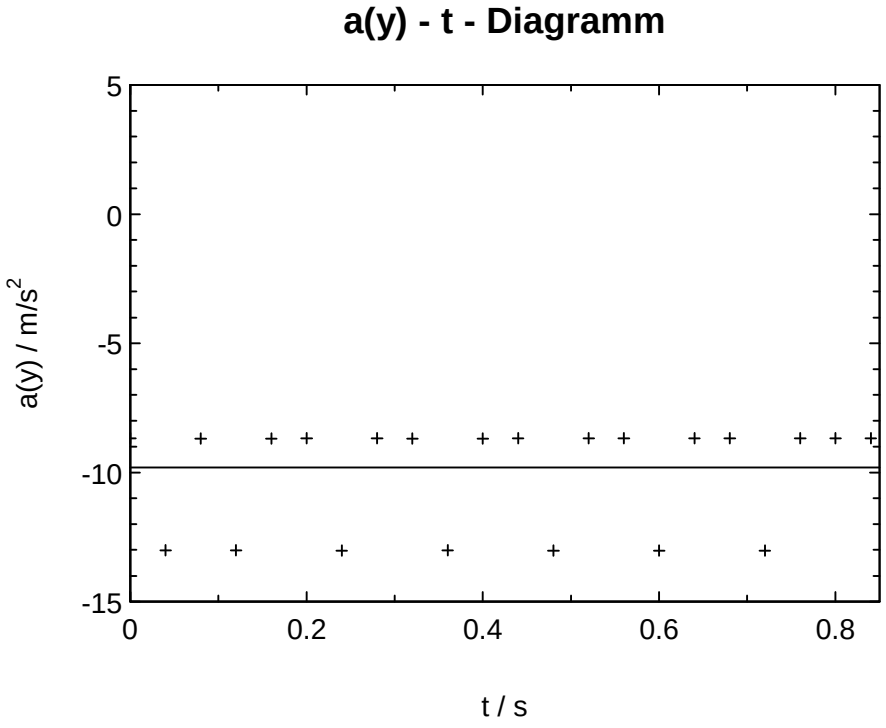
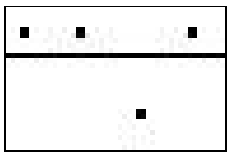


Abbildung 79

Die Beschleunigung beträgt entweder -8,6 m/s² oder -13,0 m/s². Dazwischen existieren keine Werte. Wird die Position des Objektes ein Pixel weiter oben oder unter erkannt, so springt der Beschleunigungswert gleich um 4,4 m/s² nach oben oder unten (Abbildung 79).

Die konstante Linie in dem Diagramm zeigt den tatsächlichen Wert von 9,81 m/s².

Die Werte der Beschleunigung wurden zwar korrekt erkannt, das Problem liegt vielmehr in der begrenzten Auflösung des Filmes von 288x384.[36]



In dem Beschleunigungs - Diagramm läßt sich häufig folgender Verlauf erkennen: Zwei Beschleunigungswerte wurden als -8,6 m/s² erkannt. Dadurch wurde das Objekt laut Meßwertsystem weniger beschleunigt als das in der Realität der Fall war. Durch den dritten Wert muß dies wieder ausgeglichen werden.

Während bei der Analyse des M-JPEG - Filmes offenbar keine Position falsch erkannt wurde, treten bei der Analyse des Cinepak - Filmes deutliche Fehler auf. In dem Beschleunigungs-diagramm links liegen auch Werte bei 4,2 und 17,4 m/s². Durch die Deformierung des Objektes (s. Abb.74) erkennt das Programm **Viana** die Position nicht mehr richtig. Nähert man die y-t- Werte jedoch durch eine Fit - Kurve und differenziert diese zwei mal, so erhält man g=10,2 m/s² (4% Abweichung von dem Literaturwert). Das Streuen der Meßwerte wirkt sich durch dieses Verfahren nicht bedeutend auf das Ergebnis aus.

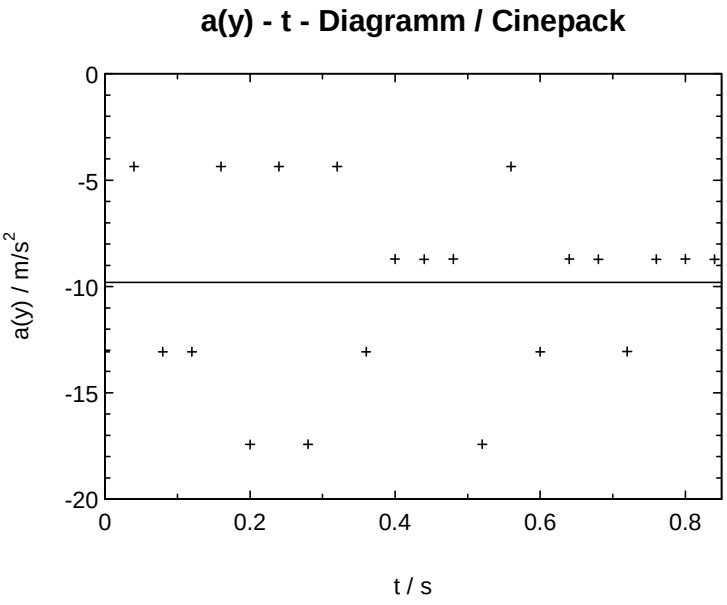


Abbildung 80

[36]Bei der höheren Auflösung von 768x576 verringert sich der Wert 4,4 auf die Hälfte. Bei der Erzeugung der Filme in dieser Auflösung traten jedoch technische Mängel bei der Digitalisierungskarte auf, die sich nach meiner Examensarbeit über den Garantieanspruch beheben lassen werden.

Beschleunigungsdiagramme wie Abbildung 80 sind im Unterricht ungeeignet, außer man thematisiert die Problematik der Auflösung.

Werden nicht die Meßwerte, sondern die Fit - Kurve abgeleitet, so ergeben sich Diagrame ohne Streuung (Abb. 81).

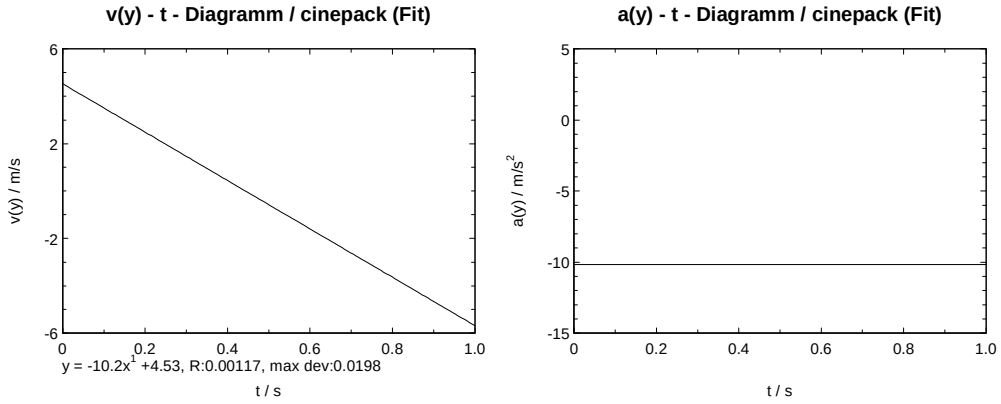


Abb. 81: Ableitungen der Fit - Kurve

Analysiert man die Bewegung des Flummis auch vor und nach der Bodenberührung (der gesamte Bewegungsablauf ist in Abbildung 68 zu sehen), so erhält man das Geschwindigkeitsdiagramm[37] in Abbildung 82 und das Beschleunigungsdiagramm in Abbildung 83.

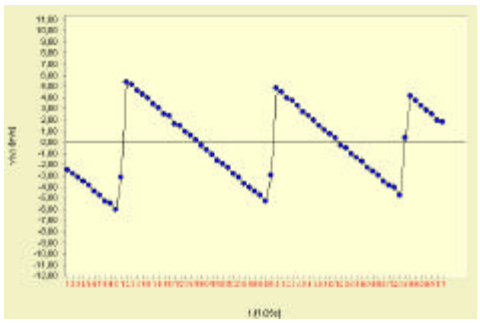


Abbildung 82: v(y) - t

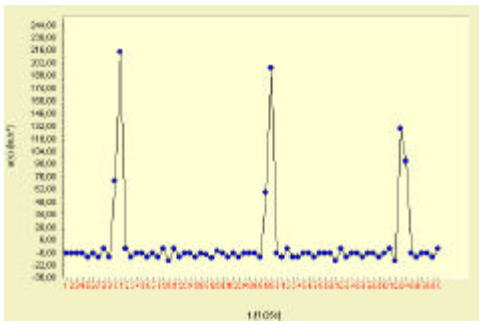


Abbildung 83: a(y) - t

[37].Während des Stoßes ändert sich die Richtung der Geschwindigkeit.

5.2 Das Federpendel

Zur Analyse einer Federschwingung wurde eine Eisenkugel rot gefärbt und von zwei Lichtquellen bestrahlt (Abbildung 84). So konnte der Schattenwurf vermieden werden. Die Schwingungsdauer beträgt etwa T = 1,3 s, so daß pro Schwingung über 30 Meßwerte entstehen. Der erzeugte Film hat eine Größe von 104 Einzelbildern. Da die Schwingung eine relativ geringe Dämpfung besitzt, sollten die Werte auf einer Sinusfunktion liegen.

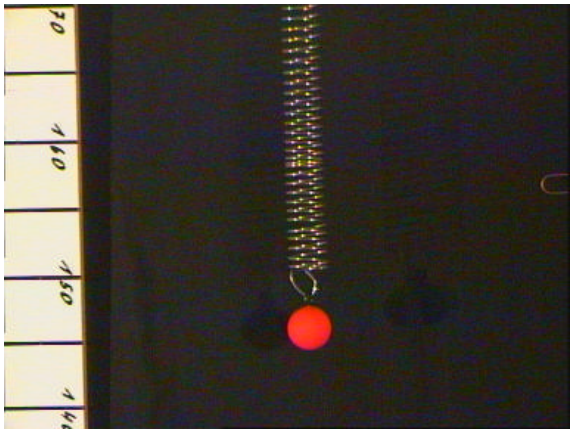


Abbildung 84

Abbildung 85 zeigt die y - t - Darstellung der gefundenen Meßwerte. Durch das Programm Easyplot wurde eine Fit - Kurve durch die Meßwerte berechnet.

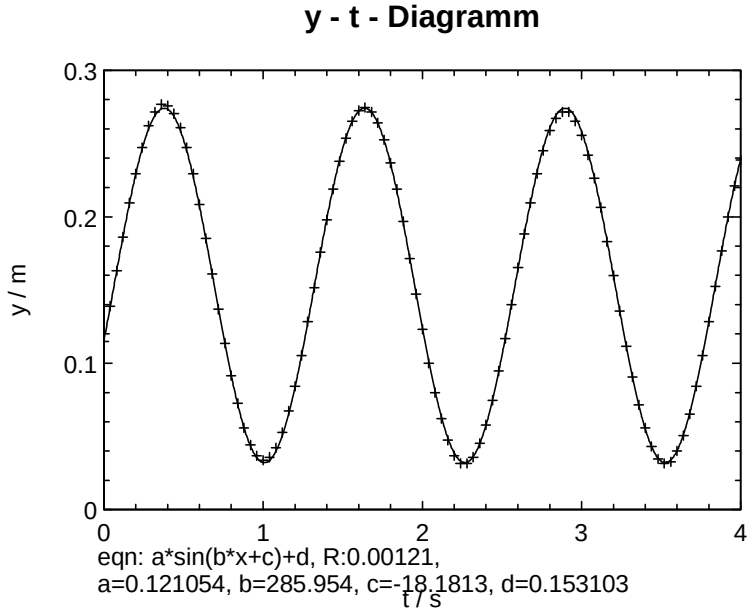
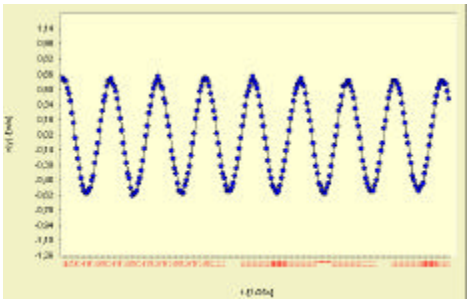


Abbildung 85

Das Geschwindigkeits- und Beschleunigungsdiagramm des Programmes **Viana** zeigt den phasenverschobenen Graphen einer Sinusfunktion.



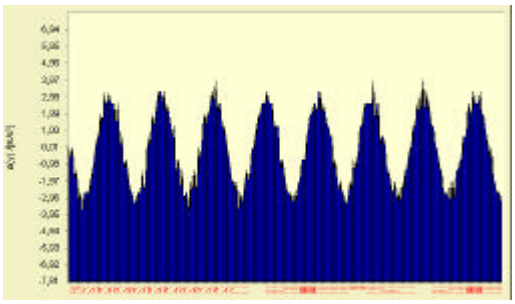


Abbildung 86

Aus der Funktionsgleichung der Fit - Funktion

$$y = 0{,}12 \cdot \sin(186 \cdot t - 18{,}2) + 0{,}15$$

läßt sich nach Umrechnung in Rad

$$y = 0{,}12 \cdot \sin(4{,}99 \cdot t - 0{,}32) + 0{,}15$$

die Frequenz $\omega = 5\ s^{-1}$ und die Amplitude A = 0,12 m ablesen.

Die Schwingung hat die Periode T = 1,26 s (oder die Frequenz $\nu = 0{,}79\ s^{-1}$).

Die Eisenkugel hat eine Masse von 110 g. Mit der Gleichung

$$\omega = \sqrt{\frac{D}{m}}$$

läßt sich so die Federkonstante D =2,75 N/m berechnen.

Um schließlich auch noch den Dämpfungsfaktor zu bestimmen, muß man eine längere Bewegung analysieren. Somit wurde ein Film mit 624 Bildern aufgenommen und analysiert. Bei der anschließenden Auswertung wurde eine der Theorie entsprechende Fit - Funktion gefunden.

$y = 0{,}125 \cdot e^{-0{,}0062 \cdot t} \cdot \sin(186 \cdot t + 161) + 0{,}15$

Die Fehlerquadratsumme zu den Meßwerten beträgt lediglich R = 0,00154.

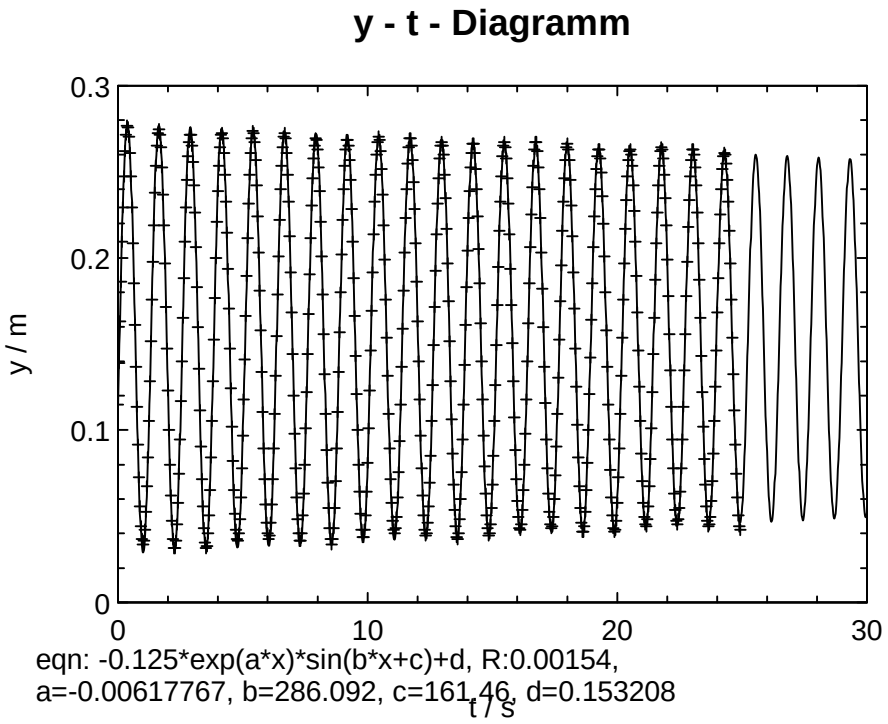


Abbildung 87

Aus der Fit - Funktion läßt sich der relativ kleine Dämpfungsfaktor $\delta = 0{,}006$ ablesen.

5.3 Die schiefe Ebene

Dieser Versuch soll zeigen, daß auch andere Darstellungen der Meßwerte sinnvoll und durchführbar sind.

Befindet sich ein Körper auf einer relativ zur Erdoberfläche schiefen Ebene, so ist die auf ihn wirkende horizontale Kraft durch die Erdanziehung abhängig vom Neigungswinkel α:

$F = \sin(\alpha) \cdot m \cdot g$

m: Masse des Körpers, g: Fallbeschleunigung

Ist die Kraft größer als die Reibungskraft, so findet eine geradlinige Bewegung mit der konstanten Beschleunigung a statt:

$a = \sin(\alpha) \cdot g \pm a_R$

a_R: Beschleunigung durch die Reibungskraft

Wählt man einen Körper (in unserem Fall einen Zylinder), der die Ebene hinabrollen kann, so hält man die Reibungskraft auf einem niedrigen Wert (Rollreibung<<Gleitreibung).

Die gewonnene Transversalgeschwindigkeit v nach dem Herunterrollen der Ebene hängt in diesem Falle jedoch noch vom Trägheitsmoment J des Körpers ab:

$E_{pot} = E_{rot} + E_{kin}$

$m \cdot g \cdot h = \frac{1}{2} J\omega^2 + \frac{1}{2} mv^2$

E_{pot}: potentielle Energie

E_{rot}: Rotationsenergie

E_{kin}: Kinetische Energie

ω: Winkelgeschwindigkeit

nach Berücksichtigen des Trägheitsmomentes $J = \frac{1}{2} mr^2$ erhält man für die Transversalgeschwindigkeit den Ausdruck:

$v = \sqrt{\frac{4}{3} g} \cdot \sqrt{h}$

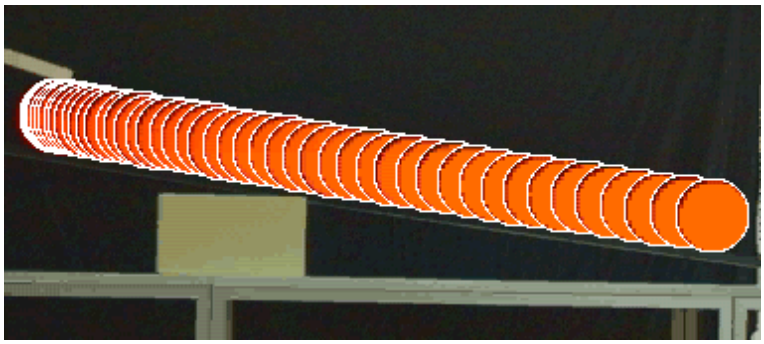


Abbildung 88

Zur Demonstration dieses Sachverhaltes wurde eine 2,7 m lange Ebene mit einer Höhendifferenz von 0,345 m aufgestellt. Die berechnete Steigung beträgt ca. 7,3° . Ob die Kamera relativ gerade aufgestellt wurde, kann man mit dem Ortsdiagramm prüfen. Ist der korrekte Winkel von Interesse, so kann man eine Ausgleichsgerade durch die Ortskurve legen (Abbildung 89). Über die Steigung läßt sich anschließend der Winkel berechnen. In unserem Beispiel beträgt dieser 8,8°. Die Kamera wurde also relativ gut justiert, prinzipiell wären aber für ähnliche Versuche Kamerastative mit eingebauter Wasserwaage sinnvoll.

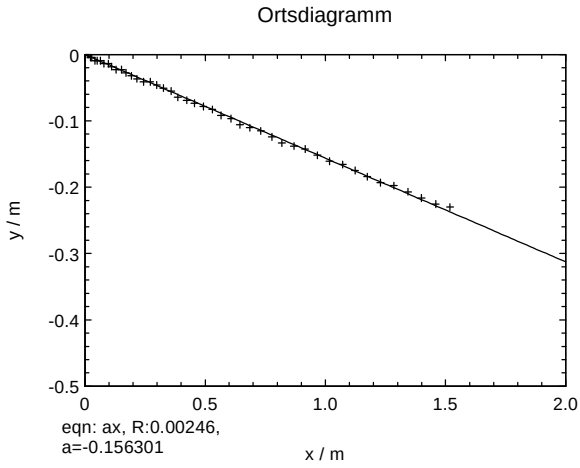


Abbildung 89

Abbildung 88 zeigt das Stroboskopbild der 49 Einzelbilder. Da sich die Farbe meiner Hand bei der gewählten Sättigungseinstellung nicht mehr ausreichend von der Objektfarbe unterschied, mußte ein Stock zum Halten des Zylinders dienen (links im Bild).

Nach der automatischen Analyse wurde die Tabelle in das Programm *Easyplot* exportiert und die Geschwindigkeit gegen die Höhe y aufgetragen (Abbildung 90). Da bei der Kalibrierung der Nullpunkt verschoben wurde, beginnen die y - Werte bei Null. Auch die Anfangsgeschwindigkeit sollte beim ersten Bild Null sein. Obwohl die Werte der kleinen Geschwindigkeiten relativ stark streuen, konnte eine Fit - Kurve durch die Meßpunkte errechnet werden:

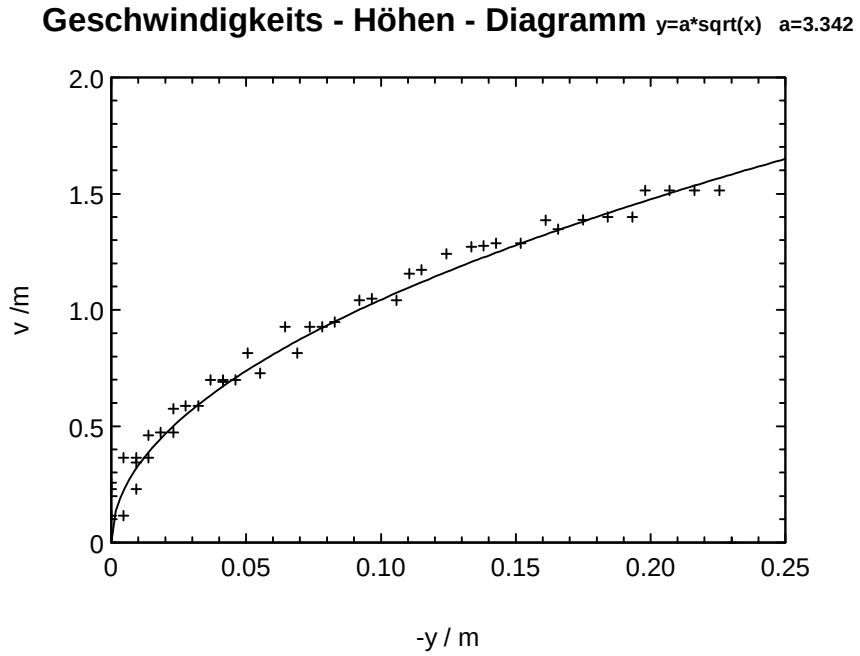


Abbildung 90

$v = 3{,}342 \cdot \sqrt{h}$

Der gefundene Faktor weist gegenüber dem theoretischen Wert $\sqrt{\frac{4}{3}g} = 3{,}62$ eine Abweichung von 7,6% auf. (Zurückzuführen auf Restreibung)

5.4 Die Verfolgung mehrerer Objekte

Bei einigen Versuchen reicht die Erfassung nur eines Objektes nicht aus. Der Zustand wäre damit nicht eindeutig beschrieben. Ein Beispiel ist der Spaceball (Abbildung 91).

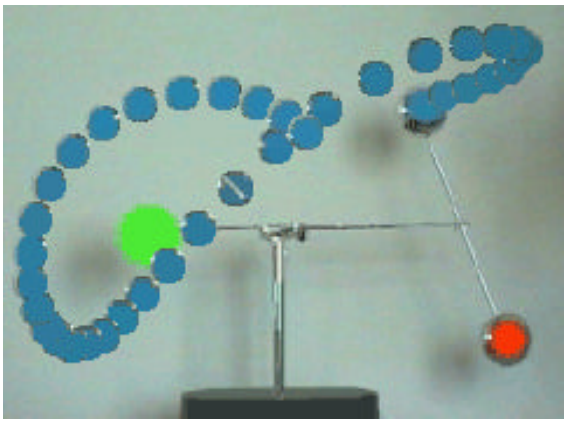


Abbildung 91

Kennt man die Position des Balles mit der blauen Markierung, so kann man allein durch diese Angabe nicht eindeutig auf die Position der Bälle mit der roten oder grünen Markierung schließen.

Wollte man bei der Videoanalyse mehrere Markierungen eines Bildes gleichzeitig erfassen, so müßten viele Einstellungen getroffen werden: Für jede Farbe eine eigene Toleranz, eine eigene Objektgröße etc.. Die Bedienung des Programmes würde noch größere Anforderungen an den Benutzer stellen.

Weitaus sinnvoller erscheint die Methode, die Objekte einzeln zu verfolgen und die gewonnenen Datenreihen später zusammenzufügen. Diese Aufgabe läßt sich mit allen Tabellenkalkulationsprogrammen oder Texteditoren bewerkstelligen.

Um dies zu demonstrieren, wurde ein Film, der die chaotische Bewegung eines Spaceballs darstellt, drei mal analysiert, die Datenreihen exportiert und anschließend in Exel zusammengefaßt. Das Ergebnis zeigt Abbildung 92.

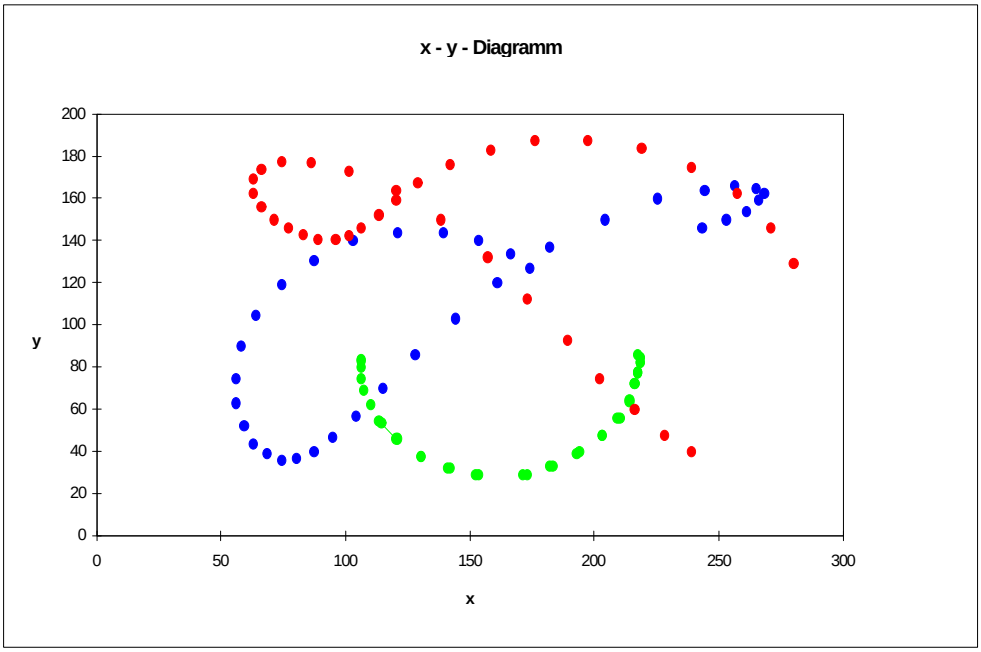


Abbildung 92

Wieviele Objekte **VIANA** noch getrennt voneinander erkennen kann, hängt im Wesentlichen von dem Geschick bei der Filmaufnahme ab. Die gewählten Farben dürfen natürlich nur wenig Ähnlichkeit miteinander besitzen.

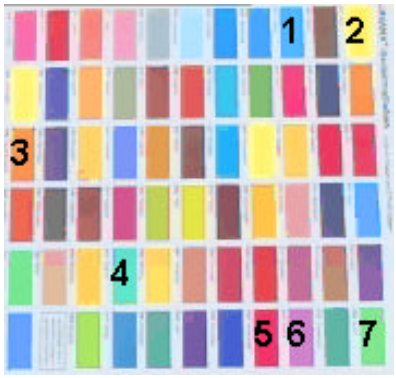


Abbildung 93

Bei der Untersuchung einer gefimlten Farbpalette wurden sieben unterscheidbare Farben gefunden (Abbildung 93). Jedes der numerierten Farbfelder wird vollständig erkannt, ohne daß **Viana** ein Pixel in den anderen sechs findet[38].

In einem Versuch könnten damit sieben Objekte verfolgt werden.

[38]Weiß, Schwarz, Grau und Braun können im Hintergrund der zu verfolgenden Objekte meist nicht vermieden werden, daher wurden sie hier nicht untersucht.

6 Schlußwort

Im Rahmen dieser Arbeit wurde das Meßwert-Erfassungssystem **VIANA** entwickelt und vorgestellt, welches sich zur Untersuchung von zweidimensionalen Bewegungen eignet. Mit Hilfe dieses Programmes lassen sich Objektpositionen automatisch bestimmen und über einen langen Zeitraum verfolgen. Die gewonnenen Koordinaten können durch das integrierte Graphikmodul in vielfältiger Weise dargestellt werden. Besonderer Wert wurde auf eine bedienbare und übersichtliche Oberfläche gelegt.

Mit **VIANA** wird die automatische Analyse einiger Versuche, die besonders reibungsarm und damit berührungsfrei ablaufen müssen, erst möglich.

Der Fehler bei der Positionsbestimmung kann bei geeigneten Einstellungen eine Bildschirmeinheit unterschreiten.

Die Kosten des Systems sind im Verhältnis zu vergleichbaren Meßwertsystemen gering. Geht man (nach [14]) davon aus, daß zukünftige Standard - Rechner Digitalisierungskarten beinhalten und außerdem eine Videokamera verfügbar ist, wird sich der Preis auf Null reduzieren.

Doch auch wenn keine Videokarte vorhanden ist, kann man das Programm nutzen. Inzwischen sind viele physikalisch relevante Digitalfilme erhältlich (s. Anhang). Das Angebot hat sich allein während der Zeit dieser Examensarbeit weiter vergrößert. Damit stehen auch Sequenzen zur Verfügung, die nicht selbst gefilmt werden können (Fallversuch auf dem Mond, Crashtest,...). Das Untersuchen solcher ‘Fremdvideos’ ist auch auf leistungsschwachen Rechnern möglich.

Mit Hilfe des Meßsystems kann sich der Physikunterricht stärker der didaktischen Forderung nach Alltagsorientierung stellen, denn es ermöglicht die Analyse zuvor gefilmter Realvorgänge oder Sequenzen aus dem Fernsehen.

Bewegungen aus verschiedensten Themengebieten lassen sich so in der Schule (teilweise fächerübergreifend) behandeln:

- Sport (Ballspiele, Leichtathletik, Extremsport, etc.)
- Verkehr
- Luftfahrt
- Wachstumsvorgänge (evtl. mit Zeitraffer)
- physikalisches Spielzeug, chaotische Bewegungen und vieles andere mehr.

Die von **VIANA** erstellten ‘Stroboskopbilder’ helfen den Schülern, den Schritt von der natürlichen Bewegung zu den abstrakten Datentabellen zu verstehen. Als ‘vertrauensbildende’ Maßnahme zu dem Meßsystem könnte der Lehrer ihnen den Ausdruck des Stroboskopbildes aushändigen und sie bitten, mit dem Lineal Meßwerte zu erfassen.
In Zukunft sollte eine Sammlung an Videosequenzen erstellt werden, die für die automatische Analyse geeignet sind. Diese Sammlung kann neben dem Programm **VIANA** über das Internet an alle Interessierten angeboten werden.
Das Programm ließe sich noch weiterentwickeln. So wäre beispielsweise eine Winkeldarstellung von sich drehenden Objekten sinnvoll. Auch das Errechnen von Fit - Kurven ließe sich in das Programm integrieren.

Literaturverzeichnis

[1] DENGLER, ROMAN u.a.,
Bewegungsabläufe mit Videokamera und Computer
In: MNU 42,5/90 S.285-290

[2] MIKELSKIS, HELMUT u.a.,
Der Computer - ein multimediales Werkzeug zum Lernen von Physik
In: Physik in der Schule 35,6/97 S.235-241

[3] MALZBENDER, HILDEGARD :
Ich schneide meine Videos selbst!
Markt & Technik 1996

[4] HAHN, HARALD :
Videobearbeitung am PC
Markt & Technik 1995

[5] EIRICH,DIETMAR :
Video für Windows Werkstatt
Markt & Technik 1995

[6] EISENKOLB, KERSTIN & WEICKARDT, HELGE :
Das Große Buch zu PC und Video
Wiener Verlag 1994

[7] KÜSTERS, HEINRICH :
Bilddatenkomprimierung mit JPEG und MPEG
Franzis Verlag 1995

[8] Miro - Video DC 30, Benutzerhandbuch
Miro Computer Products, Version 1.1/ 10.96

[9] BECK, JOSEF :
101 Festplatten im Test
In: Chip 2/97

[10] BRUNS , ENNO :
Videoschnittkarten im Test
In: Computer - Bild 4/97

[11] MEYER, EGBERT :
Videosoftware im Vergleich
In: C't 8/97, S.232 - 239

[12] GROßKOPF, RUDOLF E. :
Mustererkennung, 12. DAGM - Symposium,
Informatik - Fachberichte 254
Springer - Verlag 9/90

[13] DOBERENZ, WALTER & KOWALSKI, THOMAS :
Borland Delphi
Hanser Verlag 1995

[14] ORTLEPP, MICHAEL & HORSCH, MICHAEL :
Video für Windows
Sybex Verlag 1993

[15] EISENKOLB,KERSTIN & WEICKARDT, HELGE. :
Video PC, Desktop Video
Sybex Verlag 1995

[16] GÜTLHUBER, MICHAELA :
Handbuch zum Programm DIVA
Examensarbeit 1996,
Lehrstuhl Didaktik der Physik, Universität München

[17] SOMPLATZKI, RALF :
Mit Delphi erfolgreich programmieren
Franzis - Verlag 1995

[18] WARKEN, ELMAR :
Delphi, Entwicklung leistungsfähiger Anwendungen
Addison - Wesley 1995

[19] GERTHSEN, CHRISTIAN :
Physik, 18.Auflage
Springer - Verlag 1995

ANHANG A: Quelltext VIANA (Units 1 - 7)

```
2   unit Unit1;

4   interface

6   uses
      SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
8     Forms, Dialogs,Unit2,Unit3,Unit4,unit5,unit6,unit7, MPlayer, StdCtrls, Spin, ExtCtrls,
      clipbrd, shellapi, Buttons, Menus, Grids,chartfx;
10
    type
12  TForm1 = class(TForm)
    Film_laden: TSpeedButton;                    {Buttons}
14  ManuellBtn: TSpeedButton;
    Stroboskopbild: TButton;
16  SpeedButton2: TSpeedButton;
    AutomatischBtn: TSpeedButton;
18  abbrechenbtn: TSpeedButton;
    RadioButton1: TRadioButton;
20  RadioButton2: TRadioButton;
    AnalyseBtn: TSpeedButton;
22  EichenBtn: TSpeedButton;
    FilmanalyseBtn: TSpeedButton;
24  suchbereicheinschraenkenbtn: TSpeedButton;
    SpeedButton1: TSpeedButton;
26  DiagrammBtn: TSpeedButton;
    CheckBox1: TCheckBox;                {Optionsschalter}
28  CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
30  CheckBox5: TCheckBox;
    CheckBox2: TCheckBox;
32  CheckBox6: TCheckBox;
    CheckBox7: TCheckBox;
34  CheckBox8: TCheckBox;
    CheckBox9: TCheckBox;
36  Label1: TLabel;                              {Beschriftungen}
    Label15: TLabel;
38  Label13: TLabel;
    Label14: TLabel;
40  Label4: TLabel;
    Label6: TLabel;
42  Label7: TLabel;
    Label8: TLabel;
44  Label3: TLabel;
    Label9: TLabel;
46  Label10: TLabel;
    Label16: TLabel;
48  Label2: TLabel;
    Label17: TLabel;
50  Label5: TLabel;
    Label11: TLabel;
52  Panel1: TPanel;                              {Oberflächenfelder}
    Panel3: TPanel;
54  Panel5: TPanel;
    Panel6: TPanel;
56  Panel8: TPanel;
    Panel7: TPanel;
58  Panel9: TPanel;
    Panel10: TPanel;
60  ScrollBar1: TScrollBar                       {Laufleisten}
    ScrollBar2: TScrollBar;
62  ScrollBar3: TScrollBar;
    ScrollBar4: TScrollBar;
64  Edit1: TEdit;                                {Eingabe von Werten}
    SpinEdit2: TSpinEdit;
66  SpEToleranz: TSpinEdit;
    SpinEdit4: TSpinEdit;
68  SpinEdit3: TSpinEdit;
    SpinEdit1: TSpinEdit;
70  Edit2: TEdit;
    Edit3: TEdit;                                {Sonstiges}
72  Op end ialog1: TOp end ialog;
    Memo1: TMemo;
```

```
74      Timer1: TTimer;
        Menue: TMainMenu;
76      Datei1: TMenuItem;
        Filmladen1: TMenuItem;
78      Hilfe1: TMenuItem;
        berVIANA1: TMenuItem;
80      MediaPlayer1: TMediaPlayer;
        Image1: TImage;
82      RadioGroup2: TRadioGroup;
        RadioGroup3: TRadioGroup;
84      Extras1: TMenuItem;
        Stroboskopbild1: TMenuItem;
86      Tabelle: TStringGrid;
        savedialog1: TSaveDialog;
88                                                                  {Prozeduren, Zeile}[39]
          procedure Tabellevorbereiten;                                     1510
90        procedure Einfachesuche;                                          1316
          procedure Farbuebernahme;                                         450
92        procedure Gezieltesuche2;                                         325
          procedure gezieltesuche3(var xzentrumgenau,yzentrumgenau:integer);  691
94        procedure Bildnummer;                                             196
          procedure Zeileinsmemo(memox,memoy:integer);                      1491
96        procedure Kontextoeffnen;                                         222
          procedure farbanzeige;                                    267
98        procedure Videooeffnen;                                           201
          procedure Videoabmessungen;                                       173
100       procedure Pause(zeit:longint);                                    240
          procedure clickfarbe;                                             249
102       procedure Mittelpktzeichnen;                                      1085
          procedure Genauenmittelpktzeichnen;                               1102
104       procedure Manuellpos;                                             508
          procedure FormClose(Sender: TObject; var Action: TCloseAction);   231
106       procedure FormCreate(Sender: TObject);                            296
          procedure SuchbereicheinschraenkenBtnClick(Sender: TObject);      310
108       procedure ScrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;  493
            var ScrollPos: Integer);
110       procedure Timer1Timer(Sender: TObject);                           500
          procedure StroboskopbildClick(Sender: TObject);                   523
112       procedure CheckBox3Click(Sender: TObject);                        689
          procedure Film_ladenClick(Sender: TObject);                       155
114       procedure Filmladen1Click(Sender: TObject);                       849
          procedure ManuellBtnClick(Sender: TObject);                       868
116       procedure AutomatischBtnClick(Sender: TObject);                   882
          procedure ScrollBar2Change(Sender: TObject);                      901
118       procedure ScrollBar3Change(Sender: TObject);                      911
          procedure SpeedButton2Click(Sender: TObject);                     921
120       procedure RadioGroup2Click(Sender: TObject);                      942
          procedure CheckBox1Click(Sender: TObject);                        957
122       procedure CheckBox6Click(Sender: TObject);                        971
          procedure RadioGroup3Click(Sender: TObject);                      987
124       procedure AnalyseBtnClick(Sender: TObject);                       999
          procedure FilmanalyseBtnClick(Sender: TObject);                   1119
126       procedure AbbrechenBtnClick(Sender: TObject);             1311
          procedure EichenBtnClick(Sender: TObject);                        1422
128       procedure DiagrammBtnClick(Sender: TObject);              1530
          procedure Stroboskopbild1Click(Sender: TObject);                  523
130       procedure berVIANA1Click(Sender: TObject);                        1555
          procedure SpeedButton1Click(Sender: TObject);                     1561
132     private
          { Private-Deklarationen }
134     public
          { Public-Deklarationen }
136     end;
      var   Form1: TForm1;
138   implementation
      var Dateiname : String;
140     Abmessung : TRect;
        Farbe : tcolorref;
142     Puffer : Array[0..255] of Char;
        Fenstertitel : PChar;
144     Fensterh : THandle;
        Kontext : HDC;
146     Kontextgeoeffnet : Boolean;
```

[39] Weitere Prozeduren in Units 2-7(ab Zeile1568

```
    RSchwelle,GSchwelle,BSchwelle,toleranz:byte;
148 kreisgroesse,code,zentrumy,zentrumx,
    xzentrumgenau,yzentrumgenau:integer;
150 Playstop:boolean;
    {$R *.DFM}
152

154 procedure TForm1.Film_ladenClick(Sender: TObject);
    begin
156 screen.cursor:=crhourglass;
    If Opendialog1.Execute then Dateiname:=Opendialog1.Filename;
158 If Dateiname='' then begin                    {Falls im Dialog Abbruch gewählt wurde}
                screen.cursor:=crdefault;
160             exit;
                end;
162  Videoabmessungen;
    Videooeffnen;
164   Panel1.visible:=true;ManuellBtn.visible:=true;Film_laden.visible:=false;
      Panel6.top:=208;Panel6.left:=472;Panel6.visible:=true;
166   Mediaplayer1.width:=250;
      Label7.caption:='0';Label8.caption:=inttostr(mediaplayer1.length);
168 Timer1.enabled:=true;
    screen.cursor:=crdefault;
170 end;

172 procedure TForm1.Videoabmessungen;
                                    {X,Y-Abmessungen u. Laenge ermitteln, Ausgabe, Suchbereich=ganz}
174
     begin
176  Mediaplayer1.filename:=Dateiname;
     Mediaplayer1.Open;
178  StrPCopy(Puffer,Extractfilename(Dateiname));
     Fenstertitel := Puffer;
180  Fensterh := FindWindow('AVIWnd',Fenstertitel);
     Winprocs.GetClientRect(Fensterh,Abmessung);
182   Label4.Caption := 'Abmessungen: '+IntToStr(Abmessung.Right)+'/'+IntToStr(Abmessung.Bottom);
                                                                    {für Unit 2:}
184    Scrollbar1.Tag:=Abmessung.right;Scrollbar2.Tag:=Abmessung.bottom;
     Bildnummer;
186                                                  {Suchbereich über ganzen Bildbereich}
      Markierungx1:=0;Markierungy1:=0;
188   Markierungx2:=Abmessung.right;Markierungy2:=Abmessung.bottom;
     Scrollbar1.Max:=Mediaplayer1.length;
190  Scrollbar2.Max:=Mediaplayer1.length;
     Scrollbar3.Max:=Mediaplayer1.length;Scrollbar3.position:=mediaplayer1.length;
192  Mediaplayer1.close;
    end;
194
    procedure TForm1.Bildnummer;                    {Ausgabe der Filmposition}
196  begin
      label6.caption := Inttostr(mediaplayer1.position)+'/'+IntToStr(Mediaplayer1.length);
198  end;

200  procedure TForm1.Videooeffnen;
                                                    {Mediaplayer-Fenster öffnen}
202 begin
     If Abmessung.bottom>288 then begin             {Videogröße verkleinern}
204  Form2.clientheight:=288;
     Form2.clientwidth:=384;
206  end
     else begin                                     {Originalgröße}
208   Form2.Clientheight := Abmessung.Bottom;
      Form2.Clientwidth:=Abmessung.Right;
210  end;
     Form2.Top:=Form1.Top+45;Form2.Left:=Form1.left+5;
212   Form2.show;
     Mediaplayer1.Filename := Dateiname;
214  Mediaplayer1.Display := Form2;
     Mediaplayer1.Open;
216  Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
     Form2.refresh;
218  Mediaplayer1.Frames:=1;
    end;
220
    procedure TForm1.Kontextoeffnen;
222 begin
    StrPCopy(Puffer,'Form2');                       {Erzeugen eines PChar}
```

```
224   Fenstertitel:= Puffer;
      Fensterh := Findwindow('TForm2','Form2');          {Handle suchen lassen}
226   Kontext:= GetDC(Fensterh);                                {Kontext öffnen}
      Kontextgeoeffnet:= true;
228   end;

230   procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
      begin
232   if kontextgeoeffnet then begin              {Falls Fehler in einer Prozedur diese beendet hat}
      releasedc(fensterh,kontext);
234   messagebeep(1);                             {Hinweiston}
      end;
236   end;

238
      procedure TForm1.pause(zeit:longint);
240   var zeit1:longint;
      begin
242     zeit1:=gettickcount;
        repeat
244       application.processmessages
        until  (Gettickcount - zeit1) > zeit
246    end;

248   procedure TForm1.clickfarbe;
                              {Übergabe der angeklickten Farbe an die Variablen rschwelle,gschwelle,bschwelle}
250   var r,g,b:byte;
      begin
252   messagebeep(1);
      kontextoeffnen;
254   farbe:=getpixel(kontext,xclick,yclick);
      R:= GetRValue(Farbe);
256   G:= GetgValue(Farbe);
      B:= GetBValue(Farbe);
258    edit1.text:=inttostr(r);rschwelle:=r;
       edit2.text:=inttostr(g);gschwelle:=g;
260    edit3.text:=inttostr(b);bschwelle:=b;
      releaseDC(fensterh,kontext);                {Kontext schließen}
262   kontextgeoeffnet:=false;
      farbanzeige;                                {Prozedur zur Anzeige der drei Farbanteile}
264   end;

266   procedure TForm1.Farbanzeige;
      begin
268   with image1.canvas do begin
                                                            {Hintergrund}
270   pen.color:=clbtnface;
      brush.color:=clbtnface;
272   rectangle(0,0,image1.width,image1.height);
                                                            {Farbbalken}
274   pen.width:=1;
      pen.color:=clblack;
276   brush.color:=farbe;
      rectangle(7,7,20,88);
278                                                         {3 Balken}
      pen.width:=5;
280   pen.color:=clwindow;
      moveto(30,18);lineto(30+51,18);
282   moveto(30,50);lineto(30+51,50);
      moveto(30,79);lineto(30+51,79);
284   pen.color:=clred;
      if rschwelle>0 then begin moveto(30,18);lineto(30+(rschwelle div 5),18);end;
286
      pen.color:=clgreen;
288   if gschwelle > 0 then begin moveto(30,50);lineto(30+(gschwelle div 5),50);end;

290   pen.color:=clblue;
      if bschwelle > 0 then begin moveto(30,79);lineto(30+(bschwelle div 5),79);end;
292   end;
      end;
294
      procedure TForm1.FormCreate(Sender: TObject);
296   begin
      ManuellBtn.tag:=0;
298   Panel6.tag:=0;
      EichenBtn.Tag:=0;
300   Eichfaktor:=1;
```

```
Eifak:=1;
Memo1.text:='';
RSchwelle:=200;
GSchwelle:=100;
BSchwelle:=100;

end;

procedure TForm1.SuchbereicheinschraenkenBtnClick(Sender: TObject);
var rec : Trect;
begin
if MessageDlg('Möchten Sie den Suchbereich einschränken?Dann ziehen Sie bitte einen Markierungsrahmen auf dem Video.',
   mtInformation, [mbYes, mbNo], 0) = mrNo then exit;
SuchbereicheinschraenkenBtn.enabled:=false;
screen.cursor:=crcross;

rec.left:=form2.left;                                   {Fangen der Cursors in Form2}
rec.top:=Form2.top;
rec.right:=Form2.left+Form2.width;
rec.bottom:=Form2.top + Form2.height;
clipCursor(@rec);
end;

procedure TForm1.Gezieltesuche2;
var x2,y2,xmin,xmax,ymin,ymax : integer;
    toleranz2,xmwsumme,ymwsumme,Reihenanzahl,
    Grenzexanf,Grenzeyanf,Grenzexende,Grenzeyende:integer;
    r,g,b,fehleranzahl : byte;
    pktgefunden : boolean;
begin
Fehleranzahl:=0;
 Toleranz2:=spinedit3.value;
  xmwsumme:=0;Reihenanzahl:=0;
  Grenzeyanf:=zentrumy-(spinedit1.value div 2);                    {Objektgröße - Bereich festlegen}
    If Grenzeyanf<1 then Grenzeyanf:=1;
  Grenzeyende:=zentrumy+(spinedit1.value div 2);
    If Grenzeyende>Abmessung.bottom - 1 then Grenzeyende:=Abmessung.bottom-1;
  Grenzexanf := zentrumx -(spinedit1.value div 2);
    If Grenzexanf<1 then Grenzexanf:=1;
  Grenzexende:= zentrumx+(spinedit1.value div 2);
    If Grenzexende > Abmessung.right-1 then Grenzexende:=Abmessung.right-1;

  For y2:=Grenzeyanf  to Grenzeyende do begin
  xmin:=abmessung.right;xmax:=0;pktgefunden:=false;
  For x2 := Grenzexanf to Grenzexende do begin
   if sqr(spinedit1.value div 2) < sqr(x2-zentrumx)+sqr(y2-zentrumy) then continue;      {Kreis}
If Checkbox8.checked then form3.image1.canvas.pixels[x2,y2]:=claqua;          {Anzeige des Suchbereiches}
   Farbe:=Getpixel(Kontext,x2,y2);
     if Farbe = -1 then begin
         if fehleranzahl<20 then begin
          Form2.Bringtofront;
          Form2.refresh;
          inc(fehleranzahl);
          messagebeep(1);
         end;
        end;
   R:=GetRValue(Farbe);
   G:=getgvalue(farbe);
   B:= GetBValue(Farbe);

   if r<RSchwelle+toleranz2 then begin
   if r>rschwelle-toleranz2 then begin

    if g<GSchwelle+toleranz2 then begin
    if g>gschwelle-toleranz2 then begin

      if B<Bschwelle+toleranz2 then begin
      if b>bschwelle-toleranz2 then begin

      If Radiogroup3.Itemindex=1 then form3.image1.canvas.pixels[x2,y2]:=clblack;

       if x2 < xmin then xmin:=x2;
       if x2 > xmax then xmax:=x2;
       pktgefunden := true;
         end;
         end;
```

```
          end;
          end;
        end;
        end;
      end;
      if pktgefunden then begin
              xmwsumme:=xmwsumme + ((xmin+xmax) div 2);
              inc(Reihenanzahl);
              end;
    end;
    Pause(Scrollbar4.position);
    if Unterbrechtaste then exit;
    If Reihenanzahl<>0 then xzentrumgenau:=xmwsumme div Reihenanzahl
                    else begin
                    form3.visible:=false;
                    If abmessung.bottom>288 then form2.visible:=false;
                    if Messagedlg('Bei Zweitsuche keinen Pkt gefunden. Fehlerhafte Weierführung?',
                    mtWarning,[mbyes,mbno],0)=mrno then Unterbrechtaste:=true;
                    form3.visible:=true;
                    if abmessung.bottom>288 then begin
                                        form2.visible:=true;
                                        form2.refresh; end;
                    end;
    ymwsumme:=0;Reihenanzahl:=0;
     For x2 := Grenzexanf to Grenzexende do begin
     ymin:=abmessung.bottom;ymax:=0;pktgefunden:=false;
     For y2:= Grenzeyanf to Grenzeyende do begin
      if sqr(spinedit1.value div 2) < sqr(x2-zentrumx)+sqr(y2-zentrumy) then continue;
      Farbe:=Getpixel(Kontext,x2,y2);
         if Farbe = -1 then begin
           if fehleranzahl<20 then begin
            Form2.Bringtofront;
            Form2.refresh;
            inc(fehleranzahl);
            messagebeep(1);
           end;
          end;
        R:=GetRValue(Farbe);
        G:=getgvalue(farbe);
        B:= GetBValue(Farbe);

       if r<RSchwelle+toleranz2 then begin
       if r>rschwelle-toleranz2 then begin

         if g<GSchwelle+toleranz2 then begin
         if g>gschwelle-toleranz2 then begin

           if B<Bschwelle+toleranz2 then begin
           if b>bschwelle-toleranz2 then begin

            if y2 < ymin then ymin:=y2;
            if y2 > ymax then ymax:=y2;
            pktgefunden := true;
              end;
              end;
           end;
           end;
        end;
        end;
      end;
      if pktgefunden then begin
              ymwsumme:=ymwsumme + ((ymin+ymax) div 2);
              inc(Reihenanzahl);
              end;
    end;
    If Reihenanzahl<>0 then yzentrumgenau:=ymwsumme div Reihenanzahl;                    {s.Meldung oben}
    {zeile ins memo}
    Zeileinsmemo(xzentrumgenau,yzentrumgenau);

    If Radiogroup3.Itemindex=0 then  GenauenMittelpktzeichnen;
     end;

     procedure TForm1.Farbuebernahme;
     var r,b,g : byte;
         Altfgleichneuf : boolean;
     begin
     Altfgleichneuf:=false;
```

```
 toleranz:=SpEToleranz.Value;
456  Farbe:=Getpixel(Kontext,zentrumx,zentrumy);
   R:=GetRValue(Farbe);
458  G:=getgvalue(farbe);
   B:= GetBValue(Farbe);
460

462  if r<RSchwelle+toleranz then begin
   if r>rschwelle-toleranz then begin
464    if g<GSchwelle+toleranz then begin
    if g>gschwelle-toleranz then begin
466      if B<Bschwelle+toleranz then begin
      if b>bschwelle-toleranz then begin
468        AltFgleichneuF:=true;

470        end;
       end;
472     end;
      end;
474   end;
  end;
476 if not AltFgleichneuF then messagebeep(1) else begin
    gschwelle:=g;bschwelle:=b;rschwelle:=r;
478   end;
   edit1.text:=inttostr(rschwelle);
480  edit2.text:=inttostr(gschwelle);
   edit3.text:=inttostr(bschwelle);
482  Bildnummer;
   if altfgleichneuf then Farbanzeige;
484

486  end;

488

490

492

   procedure TForm1.ScrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;
494    var ScrollPos: Integer);
   begin
496   Label6.caption:=inttostr(scrollbar1.position)+'/'+inttostr(mediaplayer1.length);
   Mediaplayer1.position:=scrollpos;
498  end;

500  procedure TForm1.Timer1Timer(Sender: TObject);
   begin
502  If Scrollbar1.position<>mediaplayer1.position then begin
     Scrollbar1.position:=mediaplayer1.position;                                    {Laufleiste anpassen}
504    Label6.caption:=inttostr(mediaplayer1.position)+'/'+inttostr(mediaplayer1.length);   {Bildnr. ausgeben}
   end;
506  end;

508  procedure Tform1.manuellpos;
   var zeitwertstr:string;
510  begin
   Mediaplayer1.position:= Mediaplayer1.position + 1;
512  {zeile ins memo}
   str(0.2*Mediaplayer1.position:3:1,zeitwertstr);
514  memo1.lines.add(zeitwertstr+#9+inttostr(manuellclickx)+#9+inttostr(manuellclicky));
   If  Mediaplayer1.position= Mediaplayer1.length then begin
516    ManuellBtn.caption:='Manuell';
    ManuellBtn.tag:=0;
518    form2.cursor:=crdefault;
    end;
520

   end;
522

   procedure TForm1.StroboskopbildClick(Sender: TObject);
524  var x1,y1,xmin,xmax,ymin,ymax,xreihemax,xreihemin,
   fehleranzahl,bildnr,xzentrumgenau,yzentrumgenau,
526  Grenzexanf,Grenzeyanf,Grenzexende,Grenzeyende : integer;
   R,g,B : Byte;
528  AltFgleichneuF,Farbegefunden:boolean;
   begin
530  screen.cursor:=crhourglass;
   toleranz:=SpEToleranz.Value;
```

```
kreisgroesse:=0;                                        {einmaliges Festlegen der Größe}
fehleranzahl:=0;   {nur noch 10 piepser}
abbrechenbtn.visible:=true;
kontextoeffnen;
 With form3 do begin
  clientheight:=Form2.clientheight;
  clientwidth:=Form2.clientwidth;
   Image1.height:=clientheight;
   Image1.width:=clientwidth;
  show;
  end;
  PatBlt(form3.image1.canvas.handle,0,0,form3.image1.Width,
        form3.image1.Height,whiteness);                          {image löschen}
Bitblt(form3.image1.canvas.handle,0,0,form3.clientwidth,form3.clientheight,
     kontext,0,0,srccopy);                                       {videobild kopieren}
form3.refresh;
Unterbrechtaste:=false;
AltFgleichneuF:=false;

 for bildnr:=scrollbar2.position to scrollbar3.position do begin
 mediaplayer1.position:=bildnr;
 xmin:=abmessung.right;ymin:=abmessung.bottom;
 xmax:=0;ymax:=0;
  For y1:= Markierungy1 to Markierungy2 -1 do begin
  Farbegefunden:=false;xreihemax:=0;xreihemin:=abmessung.right;
  For x1 := Markierungx1 to Markierungx2 -1 do begin
   Farbe:=Getpixel(Kontext,x1,y1);
          if Farbe = -1 then begin
            If fehleranzahl<10 then begin
            inc(fehleranzahl);
            form2.bringtofront;form2.refresh;
            messagebeep(1);
            end;
           end;
   R:=GetRValue(Farbe);
   G:=getgvalue(farbe);
   B:= GetBValue(Farbe);

   if r<RSchwelle+toleranz then begin
   if r>rschwelle-toleranz then begin
     if g<GSchwelle+toleranz then begin
     if g>gschwelle-toleranz then begin
       if B<Bschwelle+toleranz then begin
       if b>bschwelle-toleranz then begin


        if x1 < xmin then xmin:=x1;
        if x1 > xmax then xmax:=x1;
        if y1 < ymin then ymin:=y1;
        if y1 > ymax then ymax:=y1;

          end;
         end;
        end;
       end;
      end;
     end;
    end;

 end;
zentrumx:=(xmin + xmax)div 2;
zentrumy :=(ymin + ymax)div 2;

gezieltesuche3(xzentrumgenau,yzentrumgenau);

    Grenzeyanf:=yzentrumgenau-(spinedit1.value div 2);
      If Grenzeyanf<1 then Grenzeyanf:=1;
    Grenzeyende:=yzentrumgenau+(spinedit1.value div 2);
      If Grenzeyende>Abmessung.bottom - 1 then Grenzeyende:=Abmessung.bottom-1;
    Grenzexanf := xzentrumgenau -(spinedit1.value div 2);
      If Grenzexanf<1 then Grenzexanf:=1;
    Grenzexende:= xzentrumgenau+(spinedit1.value div 2);
      If Grenzexende > Abmessung.right-1 then Grenzexende:=Abmessung.right-1;
                                                                {Einfarbig}
   If checkbox3.checked=true then begin
      if checkbox5.checked=true then Farbe:=Getpixel(Kontext,xzentrumgenau,yzentrumgenau)else
       farbe:=clred;                                    {Mittelpunktfarbe oder nicht}
```

```
          if checkbox4.checked=true then form3.image1.canvas.pen.color:=clwhite          {mit Rand}
610         else form3.image1.canvas.pen.color:=farbe;
          form3.image1.canvas.brush.color:=farbe;
612       form3.image1.canvas.ellipse(xzentrumgenau-(spinedit1.value div 2),
                          yzentrumgenau -(spinedit1.value div 2),
614                       xzentrumgenau+(spinedit1.value div 2),
                          yzentrumgenau +(spinedit1.value div 2));
616   end

618                          {Originalmarkierung zeichnen, Checkbox5 ist jetzt immer aus, s. chkbox3click}
    else begin
620                                                  {Weisser Kreis}
        if checkbox4.checked=true then begin
622       For y1:=Grenzeyanf  to Grenzeyende do begin
          For x1 := Grenzexanf to Grenzexende do begin
624         if sqr((spinedit1.value div 2)+1) > sqr(x1-xzentrumgenau)+sqr(y1-yzentrumgenau) then
            form3.image1.canvas.pixels[x1,y1]:=clwhite;
626       end;
          end;
628      end;
                                                              {Fuellung, etwas kleiner}
630       For y1:=Grenzeyanf  to Grenzeyende do begin
          For x1 := Grenzexanf to Grenzexende do begin
632         if sqr(spinedit1.value div 2) > sqr(x1-xzentrumgenau)+sqr(y1-yzentrumgenau) then

634         form3.image1.canvas.pixels[x1,y1]:=getpixel(kontext,x1,y1);

636       end;
          end;
638

640     end;

642                                                              {Übernahme der neuen Farbe}
  Farbe:=Getpixel(Kontext,xzentrumgenau,yzentrumgenau);
644   R:=GetRValue(Farbe);
    G:=getgvalue(farbe);
646   B:= GetBValue(Farbe);

648

    if r<RSchwelle+toleranz then begin
650   if r>rschwelle-toleranz then begin
      if g<GSchwelle+toleranz then begin
652   if g>gschwelle-toleranz then begin
        if B<Bschwelle+toleranz then begin
654     if b>bschwelle-toleranz then begin
          AltFgleichneuF:=true;
656

        end;
658    end;
      end;
660   end;
    end;
662 end;
  if not AltFgleichneuF then messagebeep(1) else begin
664   gschwelle:=g;bschwelle:=b;rschwelle:=r;
    end;
666 AltFgleichneuF:=false;
    edit1.text:=inttostr(rschwelle);
668 edit2.text:=inttostr(gschwelle);
    edit3.text:=inttostr(bschwelle);
670 Bildnummer;
     form1.refresh;
672  Farbanzeige;

674 Application.processmessages;
  if Unterbrechtaste then break;
676 end;

678 releasedc(fensterh,kontext);
  kontextgeoeffnet:=false;
680 screen.cursor:=crdefault;
  abbrechenbtn.visible:=false;
682
  end;
684
  procedure TForm1.CheckBox3Click(Sender: TObject);
```

```
686  begin
     if checkbox3.checked=false then checkbox5.enabled:=false else
688      checkbox5.enabled:=true;
     end;
690
     procedure TForm1.gezieltesuche3(var xzentrumgenau,yzentrumgenau:integer);
692  var x2,y2,xmin,xmax,ymin,ymax: integer;
       toleranz2,xmwsumme,ymwsumme,Reihenanzahl,
694    Grenzexanf,Grenzeyanf,Grenzexende,Grenzeyende:integer;
       r,g,b,fehleranzahl : byte;
696    pktgefunden : boolean;
       zeitwertstr: string;
698  begin
      Toleranz2:=spinedit2.value;
700   xmwsumme:=0;Reihenanzahl:=0;
      Grenzeyanf:=zentrumy-(spinedit1.value div 2);
702     If Grenzeyanf<1 then Grenzeyanf:=1;
      Grenzeyende:=zentrumy+(spinedit1.value div 2);
704     If Grenzeyende>Abmessung.bottom - 1 then Grenzeyende:=Abmessung.bottom-1;
      Grenzexanf := zentrumx -(spinedit1.value div 2);
706     If Grenzexanf<1 then Grenzexanf:=1;
      Grenzexende:= zentrumx+(spinedit1.value div 2);
708     If Grenzexende > Abmessung.right-1 then Grenzexende:=Abmessung.right-1;
                                                                    {x-mittelwwerte}
710   For y2:=Grenzeyanf  to Grenzeyende do begin
      xmin:=abmessung.right;xmax:=0;pktgefunden:=false;
712   For x2 := Grenzexanf to Grenzexende do begin
       if sqr(spinedit1.value div 2) < sqr(x2-zentrumx)+sqr(y2-zentrumy) then continue;
714
       Farbe:=Getpixel(Kontext,x2,y2);
716      if Farbe = -1 then begin
           if fehleranzahl<20 then begin
718         inc(fehleranzahl);
            messagebeep(1);
720        end;
          end;
722    R:=GetRValue(Farbe);
       G:=getgvalue(farbe);
724    B:= GetBValue(Farbe);

726    if r<RSchwelle+toleranz2 then begin
       if r>rschwelle-toleranz2 then begin
728
        if g<GSchwelle+toleranz2 then begin
730     if g>gschwelle-toleranz2 then begin

732      if B<Bschwelle+toleranz2 then begin
         if b>bschwelle-toleranz2 then begin
734
          if x2 < xmin then xmin:=x2;
736       if x2 > xmax then xmax:=x2;
          pktgefunden := true;
738        end;
           end;
740      end;
         end;
742    end;
       end;
744  end;
     if pktgefunden then begin
746          xmwsumme:=xmwsumme + ((xmin+xmax) div 2);
             inc(Reihenanzahl);
748          end;
750  end;
     If Reihenanzahl<>0 then xzentrumgenau:=xmwsumme div Reihenanzahl
                    else begin
752                 if Messagedlg('Bei Zweitsuche keinen Pkt gefunden. Fehlerhafte Weierführung?',
                    mtWarning,[mbyes,mbno],0)=mrno then unterbrechtaste:=true;
754                 end;
                                                                    {y-mittelwerte}
756   ymwsumme:=0;Reihenanzahl:=0;
      For x2 := Grenzexanf to Grenzexende do begin
758   ymin:=abmessung.bottom;ymax:=0;pktgefunden:=false;
      For y2:= Grenzeyanf to Grenzeyende do begin
760    if sqr(spinedit1.value div 2) < sqr(x2-zentrumx)+sqr(y2-zentrumy) then continue;
       Farbe:=Getpixel(Kontext,x2,y2);
762      if Farbe = -1 then begin
```

```
            if fehleranzahl<20 then begin
764          inc(fehleranzahl);
             messagebeep(1);
766         end;
           end;
768    R:=GetRValue(Farbe);
       G:=getgvalue(farbe);
770    B:= GetBValue(Farbe);

772    if r<RSchwelle+toleranz2 then begin
       if r>rschwelle-toleranz2 then begin
774
         if g<GSchwelle+toleranz2 then begin
776      if g>gschwelle-toleranz2 then begin

778        if B<Bschwelle+toleranz2 then begin
           if b>bschwelle-toleranz2 then begin
780
            if y2 < ymin then ymin:=y2;
782         if y2 > ymax then ymax:=y2;
            pktgefunden := true;
784           end;
              end;
786        end;
           end;
788      end;
         end;
790    end;
       if pktgefunden then begin
792           ymwsumme:=ymwsumme + ((ymin+ymax) div 2);
              inc(Reihenanzahl);
794           end;
     end;
796


798
   If Reihenanzahl<>0 then yzentrumgenau:=ymwsumme div Reihenanzahl;          {s.Meldung oben}
800
                                                                              Kreisgroesse festelegen}
802 if kreisgroesse=0 then begin
     xmin:=abmessung.right;ymin:=abmessung.bottom;
804  xmax:=0;ymax:=0;
     For x2 := Grenzexanf to Grenzexende do begin
806   For y2:= Grenzeyanf to Grenzeyende do begin
      if sqr(spinedit1.value div 2) < sqr(x2-zentrumx)+sqr(y2-zentrumy) then continue;
808   Farbe:=Getpixel(Kontext,x2,y2);
        if Farbe = -1 then begin
810       if fehleranzahl<20 then begin
           inc(fehleranzahl);
812        messagebeep(1);
          end;
814      end;
      R:=GetRValue(Farbe);
816   G:=getgvalue(farbe);
      B:= GetBValue(Farbe);
818
      if r<RSchwelle+toleranz2 then begin
820   if r>rschwelle-toleranz2 then begin

822     if g<GSchwelle+toleranz2 then begin
        if g>gschwelle-toleranz2 then begin
824
          if B<Bschwelle+toleranz2 then begin
826       if b>bschwelle-toleranz2 then begin

828        if x2 < xmin then xmin:=x2;
           if x2 > xmax then xmax:=x2;
830        if y2 < ymin then ymin:=y2;
           if y2 > ymax then ymax:=y2;
832
             end;
834          end;
           end;
836      end;
         end;
838    end;
```

```
 end;
 kreisgroesse:=((xmax-xmin+ymax-ymin) div 2)+1;
end;

 end;



procedure TForm1.Filmladen1Click(Sender: TObject);
begin
Panel7.visible:=false;
Panel8.visible:=false;
Panel5.visible:=false;
Diagrammbtn.visible:=false;
AnalyseBtn.visible:=false;
FilmanalyseBtn.visible:=false;
Eichenbtn.visible:=false;
SuchbereicheinschraenkenBtn.visible:=false;
Memo1.visible:=false;
Form3.visible:=false;
Eifak:=1;Eichfaktor:=1;
tabelle.visible:=false;
tabelle.rowcount:=2;
Form2.visible:=false;
Film_ladenclick(Form1);
end;

procedure TForm1.ManuellBtnClick(Sender: TObject);
 var pos:integer;
begin
 Mediaplayer1.display:=form7;
 form7.show;
form2.visible:=false;
 Form1.Mediaplayer1.displayrect:=Rect(0,0,Form7.clientwidth,Form7.clientheight);
 Mediaplayer1.position:=scrollbar2.position;
 nullpkty:=Form7.height;nullpktx:=0;
 Form6.show;
 Form6.bringtofront;
     Scrollbar1.Tag:=Form7.clientwidth;Scrollbar2.Tag:=Form7.clientheight;
end;

procedure TForm1.AutomatischBtnClick(Sender: TObject);
begin
Panel6.visible:=false;
Panel7.visible:=true;
Panel8.visible:=true;
Panel5.visible:=true;
Stroboskopbild1.enabled:=true;
AnalyseBtn.visible:=true;
FilmanalyseBtn.visible:=true;

DiagrammBtn.visible:=true;
Eichenbtn.visible:=true;
SuchbereicheinschraenkenBtn.visible:=true;
                                                              {für Unit 2:}
 Nullpkty:=abmessung.bottom;Nullpktx:=0;
     Scrollbar1.Tag:=Abmessung.right;Scrollbar2.Tag:=Abmessung.bottom;
     Eichfaktor:=1;
end;

procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
If scrollbar2.position>scrollbar3.position then
                scrollbar2.position:=scrollbar3.position;
Mediaplayer1.position:=scrollbar2.position;
scrollbar1.position:=scrollbar2.position;
Label7.caption:=inttostr(scrollbar2.position);
Label6.caption:=inttostr(scrollbar2.position)+'/'+inttostr(mediaplayer1.length);
end;

procedure TForm1.ScrollBar3Change(Sender: TObject);
begin
If scrollbar3.position<scrollbar2.position then
                scrollbar3.position:=scrollbar2.position;
mediaplayer1.position:=scrollbar3.position;
scrollbar1.position:=scrollbar3.position;
```

```
Label8.caption:=inttostr(scrollbar3.position);
 Label6.caption:=inttostr(scrollbar3.position)+'/'+inttostr(mediaplayer1.length);
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
playstop:=false;
speedbutton2.visible:=false;                                    {Stopbutton anzeigen}
speedbutton1.visible:=true;
mediaplayer1.position:=scrollbar2.position;                     {spule zur Anfangsposition}

repeat
application.processmessages;
mediaplayer1.position:=Mediaplayer1.position+1;
scrollbar1.position:=mediaplayer1.position;
if playstop=true then begin                                     {Falls Stopbutton gedrückt}
        speedbutton2.visible:=true;
        speedbutton1.visible:=false;
        exit;
        end;
until mediaplayer1.position=scrollbar3.position;                {Ende erreicht}
speedbutton2.visible:=true;
speedbutton1.visible:=false;
end;

procedure TForm1.RadioGroup2Click(Sender: TObject);
begin
if radiogroup2.itemindex=1 then begin                           {Einfach}
   Label10.visible:=true;
   Spinedit2.visible:=true;

   if checkbox6.checked then checkbox7.visible:=true;
  end
   else begin                                                   {Pos vorherbestimmen}
   Label10.visible:=false;
   Spinedit2.visible:=false;
   Checkbox7.visible:=false;
  end;
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
If checkbox1.checked=true then begin
    If checkbox6.checked then Checkbox8.visible:=true;
    label2.visible:=true;Spinedit3.visible:=true;
    label17.visible:=true;Spinedit1.visible:=true;
   end
   else begin
    Checkbox8.visible:=false;
    label2.visible:=false;Spinedit3.visible:=false;
    label17.visible:=false;Spinedit1.visible:=false;
   end;
end;

procedure TForm1.CheckBox6Click(Sender: TObject);
begin
If checkbox6.checked=true then begin
If Radiogroup2.Itemindex=1 then checkbox7.visible:=true;
If checkbox1.checked then checkbox8.visible:=true;
checkbox7.checked:=true;
end
else begin
checkbox7.visible:=false;
checkbox8.visible:=false;
checkbox7.checked:=false;
checkbox8.checked:=false;

end;
end;

procedure TForm1.RadioGroup3Click(Sender: TObject);
begin
if radiogroup3.itemindex=0 then begin                           {Mittelpkt}
   Radiobutton1.visible:=true;
   Radiobutton2.visible:=true;
  end
   else begin                                                   {Gef. Bildpkte}
```

```
994       Radiobutton1.visible:=false;
          Radiobutton2.visible:=false;
996     end;
        end;
998
        procedure TForm1.AnalyseBtnClick(Sender: TObject);
1000    var xMin,yMin,xMax,yMax,
            x1,y1,
1002        Fehleranzahl            : Integer;
            R,G,B :Byte;
1004    begin
        Toleranz:=SpEToleranz.value;
1006    Screen.cursor:=crHourglass;
        Fehleranzahl:=0;                                    {20 Piepser bei Fehler}
1008
         With Form3 do begin                                {Suchergebnisfenster konfigurieren,anzeigen}
1010     Clientheight:=Abmessung.Bottom;
         Clientwidth:=Abmessung.Right;
1012     Image1.Height:=Abmessung.bottom;
         Image1.Width:=Abmessung.right;
1014     Show;
         end;
1016                                                        {Suchergebnisfenster (Image1) löschen}
         PatBlt(Form3.Image1.Canvas.Handle,0,0,Form3.image1.Width,Form3.Image1.Height,Whiteness);
1018                                                        {Videofenster konfigurieren,nach oben bringen}
            Form2.Clientheight:=Abmessung.Bottom;
1020        Form2.Clientwidth:=Abmessung.Right;
            Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
1022        if Abmessung.bottom>288 then Form2.Top:=20;
            Form2.Bringtofront;
1024        form2.refresh;
        Kontextoeffnen;
1026                                                        {Suche}
        xmin:=abmessung.right;ymin:=abmessung.bottom;
1028     xmax:=0;ymax:=0;
         For y1:= Markierungy1 to Markierungy2 -1 do begin
1030     For x1 := Markierungx1 to Markierungx2 -1 do begin
          Farbe:=Getpixel(Kontext,x1,y1);
1032        if Farbe = -1 then begin                        {Farbauslesefehler}
              if fehleranzahl<20 then begin
1034           inc(fehleranzahl);
               messagebeep(1);
1036          end;
             end;
1038      R:=GetRValue(Farbe);                              {R,G,B: Farbe des ausgelesenen Pixels}
          G:=getgvalue(farbe);                              {RSchwelle,Gschwelle,Bschwelle:}
1040      B:= GetBValue(Farbe);                             {angeklickte Farbe}
1042       if R<RSchwelle+toleranz then begin
          if R>Rschwelle-toleranz then begin
1044        if G<GSchwelle+toleranz then begin
            if G>Gschwelle-toleranz then begin
1046          if B<Bschwelle+toleranz then begin
              if B>Bschwelle-toleranz then begin
1048
                  If Radiogroup3.Itemindex=1 then           {Bildpunkte}
1050                form3.image1.canvas.pixels[x1,y1]:=Farbe
                  else begin                                {Mittelpkt}
1052                  if x1 < xmin then xmin:=x1;           {kleine und größte Treffer merken}
                      if x1 > xmax then xmax:=x1;
1054                  if y1 < ymin then ymin:=y1;
                      if y1 > ymax then ymax:=y1;
1056               end;
             end;
             end;
           end;
           end;
1058      end;
          end;
1060
         end;
        end;
1062    If Radiogroup3.Itemindex=0 then begin               {Mittelpkt anzeigen}
           zentrumx:=(xmax + xMin) div 2;
1064       zentrumy:=(yMax + yMin)div 2;
           Mittelpktzeichnen;
1066       end;
```

```
    Form3.formstyle:=fsstayontop;
    Form3.bringtofront;
    Form3.refresh;
  if Abmessung.bottom>288 then begin                    {Videoanzeige verkleinern falls Übergröße}
   Form2.top:=45;
   Form2.Clientheight:=288;
   Form2.Clientwidth:=384;
   Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
  end;
 ReleaseDC(Fensterh,Kontext);
 Kontextgeoeffnet:=false;
 Screen.Cursor:=crDefault;
 end;

 procedure TForm1.Mittelpktzeichnen;
 var i:integer;
 begin
 if Radiobutton2.checked=true then                                    {.}
      Form3.Image1.Canvas.Pixels[zentrumx,zentrumy]:=clblack
      else begin                                                     {+}
  for i:=zentrumx - 2 to zentrumx + 2 do begin
    form3.image1.canvas.pixels[i,zentrumy]:=clblack;
    setpixel(kontext,i,zentrumy,16777215);
    end;
 for i:=zentrumy - 2 to zentrumy + 2 do begin
    form3.image1.canvas.pixels[zentrumx,i]:=clblack;
    setpixel(kontext,zentrumx,i,16777215);
    end;
 end;
 end;

 Procedure TForm1.GenauenMittelpktzeichnen;
 var i,j:integer;
 begin
 if Radiobutton2.checked=true then                                    {.}
      Form3.Image1.Canvas.Pixels[xzentrumgenau,yzentrumgenau]:=clblack
      else begin                                                     {+}
  for i:=xzentrumgenau - 2 to xzentrumgenau + 2 do begin
    form3.image1.canvas.pixels[i,yzentrumgenau]:=clblack;
    setpixel(kontext,i,yzentrumgenau,16777215);
    end;
 for j:=yzentrumgenau - 2 to yzentrumgenau + 2 do begin
    form3.image1.canvas.pixels[xzentrumgenau,j]:=clblack;
    setpixel(kontext,xzentrumgenau,j,16777215);
    end;
 end;
 end;

 procedure TForm1.FilmanalyseBtnClick(Sender: TObject);
 var
 x1,y1,xmin,xmax,ymin,ymax,
 bildnr,altzentrumx,altzentrumy,
 zvorausberx,zvorausbery,
 grenzexanf,grenzexende,grenzeyanf,grenzeyende : integer;
 R,g,B,fehleranzahl : Byte;
 begin
 If Radiogroup2.Itemindex=0 then begin
                    Einfachesuche;
                    exit;
                    end;
 fehleranzahl:=0;
 AbbrechenBtn.visible:=true;
 Unterbrechtaste:=false;                                          {Unterbrechung}
 toleranz:=SpEToleranz.Value;
 Tabellevorbereiten;
 memo1.clear;
 Memo1.lines.add('Bildnr.'+#9+'Zeit'+#9+'X-Koordinate'+#9+'Y-Koordinate');
 screen.cursor:=4;
  Kontextoeffnen;
                                                  {Suchergebnisfenster konfigurieren}
     With Form3 do begin
      Clientheight := Abmessung.Bottom;
      Clientwidth:=Abmessung.Right;
      Image1.height:=clientheight;
      Image1.width:=clientwidth;
      Show;
     end;
```

```
                                                    { Suchergebnisfenster (Image1) löschen}
      PatBlt(Form3.Image1.Canvas.Handle,0,0,Form3.image1.Width,Form3.Image1.Height,Whiteness);
      Form3.refresh;
                                                    {Videofenster:}
        Form2.clientheight:=Abmessung.Bottom;
        Form2.Clientwidth:=Abmessung.Right;
        Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
        if Abmessung.bottom>288 then Form2.Top:=20;
        Form2.Bringtofront;
        form2.refresh;
If Checkbox9.checked then Bitblt(form3.image1.canvas.handle,0,0,          {videobild kopieren}
                  abmessung.right,abmessung.bottom,kontext,0,0,srccopy);
                        {---------Anfangsvektor bestimmen,2xkompletten Suchbereich untersuchen,insMemo}
for bildnr:=scrollbar2.position to scrollbar2.position+1 do begin
mediaplayer1.position:=bildnr;

xmin:=abmessung.right;ymin:=abmessung.bottom;
xmax:=0;ymax:=0;
 For y1:= Markierungy1 to Markierungy2 -1 do begin
 For x1 := Markierungx1 to Markierungx2 -1 do begin
  Farbe:=Getpixel(Kontext,x1,y1);
     if Farbe = -1 then begin
       if fehleranzahl<20 then begin
        Form2.Bringtofront;
        Form2.refresh;
        inc(fehleranzahl);
        messagebeep(1);
       end;
      end;
   R:=GetRValue(Farbe);
   G:=getgvalue(farbe);
   B:= GetBValue(Farbe);

   if r<RSchwelle+toleranz then begin
   if r>rschwelle-toleranz then begin
     if g<GSchwelle+toleranz then begin
     if g>gschwelle-toleranz then begin
       if B<Bschwelle+toleranz then begin
       if b>bschwelle-toleranz then begin

         If Radiogroup3.Itemindex=1 then form3.image1.canvas.pixels[x1,y1]:=clred;

        if x1 < xmin then xmin:=x1;
        if x1 > xmax then xmax:=x1;
        if y1 < ymin then ymin:=y1;
        if y1 > ymax then ymax:=y1;
         end;
        end;
       end;
      end;
     end;
    end;
   end;
  end;
 end;
Form3.refresh;
Pause(Scrollbar4.position);
altzentrumx:=zentrumx;altzentrumy:=zentrumy;
zentrumx:=(xmin + xmax)div 2;
zentrumy :=(ymin + ymax)div 2;
If Checkbox1.checked then Gezieltesuche2;
If Radiogroup3.Itemindex=0 then begin
   If not Checkbox1.checked then Mittelpktzeichnen;
   end;

If not checkbox1.checked then                          {zeile ins memo}
     Zeileinsmemo(zentrumx,zentrumy);
  end;

{-----------------Bildnr 2 bis ende}          {letzte Bild ist Unsinn}
If Scrollbar3.position=Mediaplayer1.Length then Scrollbar3.Position:=Scrollbar3.Position -1;

for bildnr:=scrollbar2.position+2 to scrollbar3.position do begin
 mediaplayer1.position:=bildnr;

 xmin:=abmessung.right;ymin:=abmessung.bottom;
 xmax:=0;ymax:=0;
zvorausberx:=zentrumx+(zentrumx-altzentrumx);
```

```
zvorausbery:=zentrumy+(zentrumy-altzentrumy);
  Grenzeyanf:=zvorausbery-(spinedit2.value);
    If Grenzeyanf<1 then Grenzeyanf:=1;                    {Außerhalb Fenster}
  Grenzeyende:=zvorausbery+(spinedit2.value);
    If Grenzeyende>Abmessung.bottom - 1 then Grenzeyende:=Abmessung.bottom-1;
  Grenzexanf := zvorausberx -(spinedit2.value);
    If Grenzexanf<1 then Grenzexanf:=1;
  Grenzexende:= zvorausberx+(spinedit2.value);
    If Grenzexende > Abmessung.right-1 then Grenzexende:=Abmessung.right-1;

 For y1:=Grenzeyanf  to Grenzeyende do begin
  For x1 := Grenzexanf to Grenzexende do begin
  if sqr(spinedit2.value ) < sqr(x1-zvorausberx)+sqr(y1-zvorausbery) then continue;
If Checkbox7.checked then form3.image1.canvas.pixels[x1,y1]:=clyellow;{Abweichung zeichnen}
  Farbe:=Getpixel(Kontext,x1,y1);
      if Farbe = -1 then begin
        if fehleranzahl<20 then begin
         Form2.Bringtofront;
         Form2.refresh;
         inc(fehleranzahl);
         messagebeep(1);
        end;
       end;
  R:=GetRValue(Farbe);
  G:=getgvalue(farbe);
  B:= GetBValue(Farbe);

  if r<RSchwelle+toleranz then begin
  if r>rschwelle-toleranz then begin

    if g<GSchwelle+toleranz then begin
    if g>gschwelle-toleranz then begin

      if B<Bschwelle+toleranz then begin
      if b>bschwelle-toleranz then begin

       If Radiogroup3.Itemindex=1 then form3.image1.canvas.pixels[x1,y1]:=clred;

       if x1 < xmin then xmin:=x1;
       if x1 > xmax then xmax:=x1;
       if y1< ymin then ymin:=y1;
       if y1 > ymax then ymax:=y1;
         end;
         end;
       end;
       end;
    end;
    end;
  end;
 end;
Form3.refresh;
Pause(Scrollbar4.position);
If Unterbrechtaste then break;

altzentrumx:=zentrumx;altzentrumy:=zentrumy;
zentrumx:=(xmin + xmax)div 2;
zentrumy :=(ymin + ymax)div 2;

If not checkbox1.checked then                              {zeile ins memo}
    Zeileinsmemo(zentrumx,zentrumy);

if checkbox2.checked then Farbuebernahme;
If Checkbox1.checked then Gezieltesuche2;

If Radiogroup3.Itemindex=0 then begin
   If not Checkbox1.checked then Mittelpktzeichnen;
   end;

 If Unterbrechtaste then break;
 Application.Processmessages;
 If Unterbrechtaste then Break;
 end;                                                      {Scrollbar3 auf letztes Bild zurücksetzten}
If Scrollbar3.position=Mediaplayer1.Length-1 then Scrollbar3.Position:=Scrollbar3.Position +1;
 if Abmessung.bottom>288 then begin                        {Videoanzeige verkleinern falls Übergröße}
 Form2.top:=45;
 Form2.Clientheight:=288;
 Form2.Clientwidth:=384;
```

```
 Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
 end;
 ReleaseDC(Fensterh,Kontext);
 Kontextgeoeffnet:=False;
 AbbrechenBtn.visible:=false;
 screen.cursor:=crdefault;
end;


procedure TForm1.AbbrechenBtnClick(Sender: TObject);
begin
unterbrechtaste:=true;
end;

procedure TForm1.Einfachesuche;
var
x1,y1,xmin,xmax,ymin,ymax,
bildnr,
grenzexanf,grenzexende,grenzeyanf,grenzeyende : integer;
R,g,B,fehleranzahl : Byte;
begin
fehleranzahl:=0;
AbbrechenBtn.visible:=true;
Unterbrechtaste:=false;                                    {Unterbrechung}
memo1.clear;
Tabellevorbereiten;
screen.cursor:=4;
 Kontextoeffnen;
                                                           {Suchergebnisfenster konfigurieren}
    With Form3 do begin
    Clientheight := Abmessung.Bottom;
    Clientwidth:=Abmessung.Right;
    Image1.height:=clientheight;
    Image1.width:=clientwidth;
    Show;
    end;
                                                  { (Image1) löschen}
    PatBlt(Form3.Image1.Canvas.Handle,0,0,Form3.image1.Width,Form3.Image1.Height,Whiteness);
    Form3.refresh;
                                                  {Videofenster:}
  Form2.clientheight:=Abmessung.Bottom;
  Form2.Clientwidth:=Abmessung.Right;
  Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
  if Abmessung.bottom>288 then Form2.Top:=20;
  Form2.Bringtofront;
  form2.refresh;
If Checkbox9.checked then Bitblt(form3.image1.canvas.handle,0,0,    {videobild kopieren}
                         abmessung.right,abmessung.bottom,kontext,0,0,srccopy);

If Scrollbar3.position=Mediaplayer1.Length then Scrollbar3.Position:=Mediaplayer1.Length -1;
 for bildnr:=scrollbar2.position to scrollbar3.position do begin
toleranz:=SpEToleranz.Value;
 mediaplayer1.position:=bildnr;

 xmin:=abmessung.right;ymin:=abmessung.bottom;
 xmax:=0;ymax:=0;
  For y1:= Markierungy1 to Markierungy2 -1 do begin
  For x1 := Markierungx1 to Markierungx2 -1 do begin
   Farbe:=Getpixel(Kontext,x1,y1);
      if Farbe = -1 then begin
        if fehleranzahl<20 then begin
         Form2.Bringtofront;
         Form2.refresh;
         inc(fehleranzahl);
         messagebeep(1);
        end;
       end;
  R:=GetRValue(Farbe);
  G:=getgvalue(farbe);
  B:= GetBValue(Farbe);

  if r<RSchwelle+toleranz then begin
  if r>rschwelle-toleranz then begin
    if g<GSchwelle+toleranz then begin
    if g>gschwelle-toleranz then begin
      if B<Bschwelle+toleranz then begin
      if b>bschwelle-toleranz then begin
```

```
          If Radiogroup3.Itemindex=1 then form3.image1.canvas.pixels[x1,y1]:=clred;

          if x1 < xmin then xmin:=x1;
          if x1 > xmax then xmax:=x1;
          if y1 < ymin then ymin:=y1;
          if y1 > ymax then ymax:=y1;
            end;
           end;
          end;
         end;
       end;
      end;
     end;
Form3.refresh;
Pause(Scrollbar4.position);
zentrumx:=(xmin + xmax)div 2;
zentrumy :=(ymin + ymax)div 2;
if checkbox2.checked then Farbuebernahme;
If Checkbox1.checked then Gezieltesuche2;
If Radiogroup3.Itemindex=0 then begin                    {Mittelpkt zeichnen, falls nicht schon in}
   If not Checkbox1.checked then Mittelpktzeichnen;      {Gezieltesuche geschehen}
   end;

If not Checkbox1.checked then                            {zeile ins memo falls nicht schon in Gezieltes.}
    Zeileinsmemo(zentrumx,zentrumy);
if unterbrechtaste then break;
 end;

 If Scrollbar3.position=Mediaplayer1.Length-1 then Scrollbar3.Position:=Scrollbar3.Position +1;
 if Abmessung.bottom>288 then begin                      {Videoanzeige verkleinern falls Übergröße}
 Form2.top:=45;
 Form2.Clientheight:=288;
 Form2.Clientwidth:=384;
 Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
end;
ReleaseDC(Fensterh,Kontext);
Kontextgeoeffnet:=False;
AbbrechenBtn.visible:=false;
screen.cursor:=crdefault;
end;

procedure TForm1.EichenBtnClick(Sender: TObject);
var ex1,ex2,ey1,ey2:real;
begin
Panel6.Tag:=1;                                           {Kein Farbclick von unit 2}
                                  {Videofenster konfigurieren,nach oben bringen}
     Form2.Clientheight:=Abmessung.Bottom;
     Form2.Clientwidth:=Abmessung.Right;
     Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
     if Abmessung.bottom>288 then Form2.Top:=20;
     Form2.Bringtofront;
     form2.refresh;
MessageDlg('Bitte clicken Sie auf den Anfangspunkt einer Strecke bekannten Ausmaßes.',
       mtinformation,[mbOK],0);
Screen.cursor:=6;
EichenBtn.Tag:=1;                                        {Flagge für Unit2 ,1.Eichungswert}
Eichungx1:=-1;                                           {Hilfswert für repeat-Bedingung}
Repeat
Application.processmessages;
until Eichungx1<>-1;
Screen.Cursor:=crdefault;

MessageDlg('Bitte clicken Sie jetzt auf den Endpunkt.',
       mtinformation,[mbOK],0);
Screen.cursor:=6;
EichenBtn.Tag:=2;                                {Flagge für Unit2 ,2.Eichungswert}
Eichungx2:=-1;
Repeat
Application.processmessages;
until Eichungx2<>-1;
Screen.Cursor:=crdefault;
EichenBtn.Tag:=0;

Form4.show;
Form4.bringtofront;
```

```
1456    Streckeinm:=0;
        Repeat
1458    Application.processmessages;
        until Streckeinm<>0;
1460    Form4.visible:=false;

1462    ex1:=eichungx1;ex2:=eichungx2;ey1:=eichungy1;ey2:=eichungy2;
        try
1464      Eichfaktor:=Streckeinm/Sqrt(Sqr(Ex2-Ex1)+Sqr(Ey2-Ey1));
        except
1466     Messagedlg('Fehler.'+#13+'Bitte wiederholen Sie die Eichung.',mtinformation,[mbok],0);
        end;
1468
        Eifak:=Eichfaktor;
1470    if MessageDlg('Möchten Sie noch den Nullpunkt verschieben?',
              mtinformation,[mbyes,mbno],0)=mryes then begin
1472        Screen.cursor:=6;
            EichenBtn.Tag:=3;                          {Flagge für Unit2 ,Nullpkt}
1474        Nullpktx:=-1;
            Repeat
1476        Application.processmessages;
            if unterbrechtaste then break;
1478        until Nullpktx<>-1;
            Screen.Cursor:=crdefault;
1480        EichenBtn.Tag:=0;
            end;
1482
        if Abmessung.bottom>288 then begin          {Videoanzeige verkleinern falls Übergröße}
1484     Form2.top:=45;
         Form2.Clientheight:=288;
1486     Form2.Clientwidth:=384;
         Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
1488    end;
        Panel6.tag:=0;
1490    end;
        procedure TForm1.Zeileinsmemo(memox,memoy:integer);
1492    var Zeitwert,xKoordinate,YKoordinate:real;
          Zeitwertstr,XKoordinatestr,YKoordinatestr:string;
1494    begin
        Zeitwert:=(Mediaplayer1.position-scrollbar2.position)/Spinedit4.value;
1496        str(Zeitwert:3:3,zeitwertstr);
        xKoordinate:=(memox-Nullpktx)*Eichfaktor;
1498        Str(xKoordinate:5:5,XKoordinatestr);
        YKoordinate:=(Nullpkty-memoy)*Eichfaktor;
1500        Str(YKoordinate:5:5,Ykoordinatestr);
            memo1.lines.add(Zeitwertstr+#9+xkoordinatestr+#9+YKoordinatestr);
1502    Tabelle.rowcount:=Mediaplayer1.position-scrollbar2.position+2;
            tabelle.cells[0,Mediaplayer1.position-scrollbar2.position+1]:=zeitwertstr;
1504        Tabelle.cells[1,Mediaplayer1.position-scrollbar2.position+1]:=XKoordinatestr;
            Tabelle.cells[2,Mediaplayer1.position-scrollbar2.position+1]:=YKoordinatestr;
1506    end;
1508

1510    procedure TForm1.Tabellevorbereiten;
        var i:integer;
1512    begin

1514    Tabelle.color:=16777172;
        Tabelle.Fixedcolor:=14671752;
1516    For i:=0 to Tabelle.Rowcount do begin                              {Löschen}
        Tabelle.cells[0,i]:='';
1518    Tabelle.cells[1,i]:='';
        Tabelle.cells[2,i]:='';
1520    end;
        Tabelle.rowcount:=2;
1522    tabelle.cells[0,0]:='Zeit/s';
        tabelle.cells[1,0]:=' X/m';
1524    tabelle.cells[2,0]:=' Y/m';
        Tabelle.visible:=true;
1526
        end;
1528

1530    procedure TForm1.DiagrammBtnClick(Sender: TObject);
        begin
1532    if tabelle.rowcount<4 then exit;
```

```
form5.show;
form5.bringtofront;
form5.XYDiagr;
form5.tabellefuellen;
end;

procedure TForm1.Stroboskopbild1Click(Sender: TObject);
begin
if Stroboskopbild1.checked=false then begin
Stroboskopbild1.checked:=true;
Panel3.top:=480;Panel3.left:=384;
Panel3.visible:=true;
Diagrammbtn.visible:=false;
tabelle.visible:=false;
end
else begin
Panel3.visible:=false;
Diagrammbtn.visible:=true;
stroboskopbild1.checked:=false;
end;
end;

procedure TForm1.berVIANA1Click(Sender: TObject);
begin
messagedlg('Automatische VIdeoANAlyse'+#13+#10+'von Thomas Kersting'+#10+#13+
        'Uni-Essen, Didaktik der Physik',mtinformation,[mbok],0);
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
playstop:=true;
end;

end.
```

```
unit Unit2;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms,unit3, Dialogs;

type
  TForm2 = class(TForm)
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormCreate(Sender: TObject);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormPaint(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form2: TForm2;
  markierungx1,markierungy1,
  markierungx2,markierungy2,
  xclick,yclick,manuellclickx,manuellclicky,
  Eichungx1,Eichungx2,Eichungy1,Eichungy2,
  Nullpktx,Nullpkty:integer;
  maustaste : boolean;
implementation

{$R *.DFM}

uses unit1;
const creichen=1;
procedure TForm2.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
if Button=mbRight then form1.mediaplayer1.position:=          {Bei rechter Maustaste ein Bild weiterspulen}
   form1.mediaplayer1.position + 1 else begin
                                                {Suchbereicheinschränken,ersten Markierungsrahmen zeichnen}
   if form1.suchbereicheinschraenkenBtn.enabled=false then begin
      markierungx1:=x;markierungx2:=x;
      markierungy1:=y;markierungy2:=y;
      canvas.pen.mode:=pmnotxor;
      canvas.pen.style:=psdash;
      canvas.brush.style:=bsclear;
      canvas.rectangle(x,y,x,y);
      maustaste:=true;
      exit;
      end;

  if form1.manuellBtn.tag=1 then begin
   manuellclickx:=x;
   manuellclicky:=y;
   form1.manuellpos;
   exit;
  end;
                                        {Kalibrieren des Bildschirms, 1: Anfangspunkt, 2:Endpunkt, 3:Nullpkt}
 If Form1.EichenBtn.Tag=1 then begin
  Eichungx1:=x;
  Eichungy1:=y;
  end;
  If Form1.EichenBtn.Tag=2 then begin
  Eichungx2:=x;
  Eichungy2:=y;
  end;
  If Form1.EichenBtn.Tag=3 then begin
  Nullpktx:=x;
  Nullpkty:=y;
  end;
```

```
                                                    {Farbe auslesen}
If Form1.Panel6.tag=0 then begin
      xclick:=x;
      yclick:=y;
      form1.clickfarbe;
      end;
end;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
maustaste:=false;

end;

procedure TForm2.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
if not maustaste then exit;
canvas.rectangle(markierungx1,markierungy1,markierungx2,markierungy2);
markierungx2:=x;markierungy2:=y;
canvas.rectangle(markierungx1,markierungy1,markierungx2,markierungy2);
end;

procedure TForm2.FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
if Form1.suchbereicheinschraenkenBtn.enabled = true then exit;
maustaste:=false;
screen.cursor:=crdefault;
Form1.SuchbereicheinschraenkenBtn.enabled:=true;
canvas.rectangle(markierungx1,markierungy1,markierungx2,markierungy2);
if Markierungx1>Markierungx2 then begin
                  x:=Markierungx1;
                  Markierungx1:=Markierungx2;
                  Markierungx2:=x;
                  end;
 if Markierungy1>Markierungy2 then begin
                  y:=Markierungy1;
                  Markierungy1:=Markierungy2;
                  Markierungy2:=y;
                  end;
clipcursor(nil);
 If Form1.Scrollbar2.Tag>288 then begin
 messagebeep(1);
 Markierungx1:=Markierungx1*Form1.Scrollbar1.Tag div 384;
 Markierungx2:=Markierungx2*Form1.Scrollbar1.Tag div 384;
 Markierungy1:=Markierungy1*Form1.Scrollbar2.Tag div 288;
 Markierungy2:=Markierungy2*Form1.Scrollbar2.Tag div 288;

 end;
if MessageDlg('Horizontal: '+inttostr(markierungx1)+' - '+inttostr(markierungx2)
+#10+#13+'Vertikal: '+inttostr(markierungy1)+' - '+inttostr(markierungy2),
   mtInformation, [mbYes, mbNo], 0) = mrYes then exit
   else begin
     markierungx1:=0;markierungy1:=0;
    markierungx2:=clientwidth;markierungy2:=clientheight;
MessageDlg('Abbruch: Das Programm untersucht den ganzen Bildbereich',
   mtInformation, [mbOK], 0)
    end;
end;

procedure TForm2.FormPaint(Sender: TObject);          {Anzeigen des Videos bei Wiedereinblenden}
var pos:integer;
begin
pos := Form1.Mediaplayer1.position;
with Form1.Mediaplayer1 do begin
play;
position:=pos;
end;

end;

procedure TForm2.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
unterbrechtaste:=true;
```

```
end;
end.
{-------------------------------------------------------------------------------------------------------------------}
unit Unit3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, ExtCtrls;

type
  TForm3 = class(TForm)
    Image1: TImage;
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form3: TForm3;
   Unterbrechtaste:boolean;
implementation
{$R *.DFM}

procedure TForm3.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
Unterbrechtaste:=true;
end;
end.
```

```
unit 4;
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons;

type
  TForm4 = class(TForm)
    BitBtn1: TBitBtn;
    Edit1: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label5: TLabel;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form4: TForm4;
  Streckeinm:real;
implementation

{$R *.DFM}

procedure TForm4.BitBtn1Click(Sender: TObject);
var code:integer;

begin
val(edit1.text,Streckeinm,code);
end;

end.
```

```
unit Unit5;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms,clipbrd, Dialogs, VBXCtrl, Chart2fx, StdCtrls, ExtCtrls, Buttons, Grids;

type
  TForm5 = class(TForm)
    diagramm: TChartFX;
    Groupbox1: TGroupBox;
    yx: TRadioButton;
    Xt: TRadioButton;
    yt: TRadioButton;
    vt: TRadioButton;
    vxt: TRadioButton;
    vyt: TRadioButton;
    at: TRadioButton;
    axt: TRadioButton;
    ayt: TRadioButton;
    SpeedButton1: TSpeedButton;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Bevel3: TBevel;
    tabelle2: TStringGrid;
    Panel1: TPanel;
    SpeedButton2: TSpeedButton;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    procedure XYDiagr;
    procedure tabellefuellen;
    procedure bereich(wert:real);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Radioclick(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form5: TForm5;
  eifak:real;
implementation
   uses chartFX,unit1,unit2;
{$R *.DFM}
var maxiy,miniy:real;

procedure TForm5.xyDiagr;
var i,code:integer;
    Wert:real;
begin
diagramm.rgb2dbk:=13959167;
diagramm.rgbbk:=12972786;
yx.checked:=true;
diagramm.custtool:=332595200;
diagramm.title[CHART_bottomTit]:='x /m';
diagramm.title[CHART_leftTit]:='y /m';
diagramm.opendata[cod_xvalues]:=makelong(1,Form1.Tabelle.rowcount-1);
diagramm.thisserie:=0;
For i:=1 to Form1.Tabelle.rowcount-1 do begin
val(form1.tabelle.cells[1,i],wert,code);
bereich(wert);
Diagramm.xvalue[i-1]:=wert;
end;
diagramm.closedata[cod_xvalues]:=0;

diagramm.adm[csa_max]:=nullpkty*Eifak;
diagramm.adm[csa_min]:=(nullpkty-Form1.scrollbar2.tag)*eifak;
diagramm.adm[csa_xmax]:=(Form1.scrollbar1.tag-nullpktx)*Eifak;
diagramm.adm[csa_xmin]:=-nullpktx*Eifak;

 diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-1);
 diagramm.thisserie:=0;
 For i:=1 to Form1.Tabelle.rowcount-1 do begin
```

```
1868    val(form1.tabelle.cells[2,i],wert,code);
        Diagramm.value[i-1]:=wert;
1870    end;
        Form5.diagramm.closedata[cod_values]:=0;
1872    Diagramm.charttype:=8;
        end;
1874
        procedure TForm5.SpeedButton1Click(Sender: TObject);
1876    begin
        form5.close;
1878    end;
        procedure TForm5.bereich(wert:real);
1880    begin
        if miniy>wert then miniy:=wert;
1882    if maxiy<wert then maxiy:=wert;
        end;
1884
        procedure TForm5.Radioclick(Sender: TObject);
1886    var i,code:integer;
           wert,wert1,wert2:real;
1888       s:string;
        begin
1890    if Sender=yx then begin
                        xydiagr;
1892                    exit;
                        end;
1894    miniy:=30000;maxiy:=-30000;
1896
        if Sender =xt then begin
1898        diagramm.title[CHART_leftTit]:='x /[m]';
           diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-1);
1900       diagramm.thisserie:=0;
           For i:=1 to Form1.Tabelle.rowcount-1 do begin
1902       val(form1.tabelle.cells[1,i],wert,code);
           bereich(wert);
1904       Diagramm.value[i-1]:=wert;
           end;
1906       diagramm.closedata[cod_values]:=0;
          end;
1908     if Sender =yt then begin
            diagramm.title[CHART_leftTit]:='y /[m]';
1910       diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-1);
           diagramm.thisserie:=0;
1912       For i:=1 to Form1.Tabelle.rowcount-1 do begin
           val(form1.tabelle.cells[2,i],wert,code);
1914       bereich(wert);
           Diagramm.value[i-1]:=wert;
1916       end;
           diagramm.closedata[cod_values]:=0;
1918      end;
1920
1922    if Sender= vxt then begin
            diagramm.title[CHART_leftTit]:='v(x) /[m/s]';
1924       diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-2);
           diagramm.thisserie:=0;
1926       For i:=1 to Form1.Tabelle.rowcount-2 do begin
           val(tabelle2.cells[5,i],wert,code);
1928       bereich(wert);
           Diagramm.value[i-1]:=wert;
1930       end;
           diagramm.closedata[cod_values]:=0;
1932       end;
1934    if Sender=vyt then begin
            diagramm.title[CHART_leftTit]:='v(y) /[m/s]';
1936       diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-2);
           diagramm.thisserie:=0;
1938       For i:=1 to Form1.Tabelle.rowcount-2 do begin
           val(tabelle2.cells[6,i],wert,code);
1940       bereich(wert);
           Diagramm.value[i-1]:=wert;
1942       end;
           diagramm.closedata[cod_values]:=0;
1944       end;
```

```
1946    if Sender=vt then begin
          diagramm.title[CHART_leftTit]:='v /[m/s]';
1948      diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-2);
          diagramm.thisserie:=0;
1950      For i:=1 to Form1.Tabelle.rowcount-2 do begin
          val(tabelle2.cells[4,i],wert,code);
1952      bereich(wert);
          Diagramm.value[i-1]:=wert;
1954      end;
          diagramm.closedata[cod_values]:=0;
1956      end;

1958

1960    if Sender=ayt then begin
          if tabelle2.rowcount<5 then exit;
1962      diagramm.title[CHART_leftTit]:='a(y) /[m/s²]';
          diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-3);
1964      diagramm.thisserie:=0;
          For i:=1 to Form1.Tabelle.rowcount-3 do begin
1966      val(tabelle2.cells[9,i],wert,code);
          bereich(wert);
1968      Diagramm.value[i-1]:=wert;
          end;
1970      diagramm.closedata[cod_values]:=0;
          end;
1972    if Sender=axt then begin
          if tabelle2.rowcount<5 then exit;
1974       diagramm.title[CHART_leftTit]:='a(x) /[m/s²]';
          diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-3);
1976      diagramm.thisserie:=0;
          For i:=1 to Form1.Tabelle.rowcount-3 do begin
1978      val(tabelle2.cells[8,i],wert,code);
          bereich(wert);
1980      Diagramm.value[i-1]:=wert;
          end;
1982      diagramm.closedata[cod_values]:=0;
          end;
1984    if Sender=at then begin
          if tabelle2.rowcount<5 then exit;
1986      diagramm.title[CHART_leftTit]:='a /[m/s²]';
          diagramm.opendata[cod_values]:=makelong(1,Form1.Tabelle.rowcount-3);
1988      diagramm.thisserie:=0;
          For i:=1 to Form1.Tabelle.rowcount-3 do begin
1990      val(tabelle2.cells[7,i],wert,code);
          bereich(wert);
1992      Diagramm.value[i-1]:=wert;
          end;
1994      diagramm.closedata[cod_values]:=0;
          end;
1996    if tabelle2.rowcount>25 then diagramm.fixedgap:=90 div tabelle2.rowcount;
        If tabelle2.rowcount>89 then diagramm.fixedgap:=2;
1998    diagramm.title[CHART_bottomTit]:='t /[1/'+inttostr(form1.spinedit4.value)+'s]';
        Diagramm.charttype:=1;
2000    diagramm.custtool:=332601152;
        if maxiy >=0 then diagramm.adm[csa_max]:=maxiy*2 else diagramm.adm[csa_max]:=maxiy/2;
2002    if miniy>=0 then diagramm.adm[csa_min]:=miniy/2 else diagramm.adm[csa_min]:=miniy*2;
        end;
2004
        procedure Tform5.tabellefuellen;
2006    var i,j,code:integer;
          wert,wert1,wert2,wert3:real;
2008      wertstr:string;
        begin
2010    with tabelle2 do begin
        color:=16777172;
2012    Fixedcolor:=14671752;
        rowcount:=Form1.tabelle.rowcount;
2014    cells[0,0]:='Bildnr.';
        cells[1,0]:='t /s';
2016    cells[2,0]:='x /m';
        cells[3,0]:='y /m';
2018    cells[4,0]:='v /m/s';
        cells[5,0]:='v(x) /m/s';
2020    cells[6,0]:='v(y) /m/s';
        cells[7,0]:='a /m/s²';
```

```
2022    cells[8,0]:='a(x) /m/s²';
        cells[9,0]:='a(y) /m/s²';
2024    end;
        for i:=1 to Form1.Tabelle.rowcount-1 do begin
2026      for j:=0 to 2 do begin
           tabelle2.cells[j+1,i]:=Form1.tabelle.cells[j,i];
2028       end;
         tabelle2.cells[0,i]:=inttostr(i);
2030     end;
        For i:=1 to Tabelle2.rowcount-2 do begin                    {vy}
2032      val(tabelle2.cells[3,i],wert1,code);
          val(tabelle2.cells[3,i+1],wert2,code);
2034      wert:=(wert2-wert1)*Form1.Spinedit4.value;
          Str(wert:5:5,wertstr);
2036      tabelle2.cells[6,i]:=wertstr;
          end;
2038    For i:=1 to Tabelle2.rowcount-2 do begin                    {vx}
          val(tabelle2.cells[2,i],wert1,code);
2040      val(tabelle2.cells[2,i+1],wert2,code);
          wert:=(wert2-wert1)*Form1.Spinedit4.value;
2042      Str(wert:5:5,wertstr);
          tabelle2.cells[5,i]:=wertstr;
2044      end;
        For i:=1 to Tabelle2.rowcount-2 do begin                     {v}
2046      val(tabelle2.cells[5,i],wert1,code);
          val(tabelle2.cells[6,i],wert2,code);
2048      wert:=Sqrt(Sqr(wert1)+Sqr(wert2));
          Str(wert:5:5,wertstr);
2050      tabelle2.cells[4,i]:=wertstr;
          end;
2052    For i:=1 to Tabelle2.rowcount-3 do begin                     {ay}
          val(tabelle2.cells[3,i],wert1,code);
2054      val(tabelle2.cells[3,i+1],wert2,code);
          val(tabelle2.cells[3,i+2],wert3,code);
2056      wert:=(wert3-(2*wert2)+wert1)*sqr(form1.spinedit4.value);
          Str(wert:5:5,wertstr);
2058      tabelle2.cells[9,i]:=wertstr;
          end;
2060     For i:=1 to Tabelle2.rowcount-3 do begin                              {ax}
          val(tabelle2.cells[2,i],wert1,code);
2062      val(tabelle2.cells[2,i+1],wert2,code);
          val(tabelle2.cells[2,i+2],wert3,code);
2064      wert:=(wert3-(2*wert2)+wert1)*sqr(form1.spinedit4.value);
          Str(wert:5:5,wertstr);
2066      tabelle2.cells[8,i]:=wertstr;
          end;
2068     For i:=1 to Tabelle2.rowcount-3 do begin                              {a}
          val(tabelle2.cells[8,i],wert1,code);
2070      val(tabelle2.cells[9,i],wert2,code);
          wert:=Sqrt(Sqr(wert1)+Sqr(wert2));
2072      Str(wert:5:5,wertstr);
          tabelle2.cells[7,i]:=wertstr;
2074      end;

2076
        For i:=4 to 9 do begin                                  {Einträge von voher löschen}
2078    tabelle2.cells[i,tabelle2.rowcount-1]:='';
        end;
2080    for i:=7 to 9 do begin
        tabelle2.cells[i,tabelle2.rowcount-2]:='';
2082    end;
        end;
2084
        procedure TForm5.SpeedButton2Click(Sender: TObject);
2086    var i,j,posi,size:integer;
          buffer:pchar;
2088      Datname,s,wstr,komma:string;
        begin
2090    if radiobutton1.checked then komma:=',' else komma:='.';
        Form1.memo1.clear;
2092    Form1.memo1.lines.add('t/[s]'+#9+'x/[m]'+#9+'y/[m]'+#9+'v/[m/s]'+#9+'v(x)/[m/s]'+#9+'v(y)[m/s]'+
                    #9+'a[m/s²]'+#9+'a(x)[m/s²]'+#9+'a(y)[m/s²]');
2094    for i:=1 to tabelle2.rowcount-1 do begin
        wstr:='';
2096     for j:=1 to 9 do begin
          s:=tabelle2.cells[j,i];
2098      posi:=pos('.',s);
```

```
      delete(s,posi,1);
2100  insert(komma,s,posi);
     wstr:=wstr+#9+s;
2102  end;
     delete(wstr,1,1);
2104  form1.memo1.lines.add(wstr); end;
     size := Form1.memo1.gettextlen;                                    {Zeichenanzahl}
2106  inc(size);                                                         {Platz für Nullzeichen}
     getmem(buffer,size);                                               {dyn. Speicher}
2108  Form1.memo1.gettextbuf(buffer,size);                               {kopiert in buffer}
     clipboard.settextbuf(buffer);                                      {ab in Zwischenabl}
2110  freemem(buffer,size);
     if Messagedlg('Soll die Tabelle außer in der Zwischenablage noch in einer Datei gespeichert
2112 werden?',mtconfirmation,
              [mbyes,mbno],0)=mrno then exit;
2114 datname:='';
     if Form1.savedialog1.execute then Datname:=Form1.saveDialog1.Filename;
2116
     try Form1.memo1.lines.savetofile(datname);
2118   except
         end;
2120   end;

2122 end.
     {-------------------------------------------------------------------------------------------------------------------------------}
2124 unit Unit6;

2126 interface
     uses
2128   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
      Forms, Dialogs, StdCtrls, ExtCtrls, Buttons;
2130
     type
2132   TForm6 = class(TForm)
        Bevel1: TBevel;
2134    Label1: TLabel;
        Label2: TLabel;
2136    Bevel2: TBevel;
        Label3: TLabel;
2138    Label4: TLabel;
        Label5: TLabel;
2140    Label6: TLabel;
        DiagrammBtn: TSpeedButton;
2142    kalibrierenbtn: TSpeedButton;
        SpeedButton2: TSpeedButton;
2144    BitBtn1: TBitBtn;
        Image1: TImage;
2146    Image2: TImage;
        Image3: TImage;
2148    Label7: TLabel;
        procedure FormClose(Sender: TObject; var Action: TCloseAction);
2150    procedure kalibrierenbtnClick(Sender: TObject);
        procedure DiagrammBtnClick(Sender: TObject);
2152    procedure SpeedButton2Click(Sender: TObject);
        procedure FormPaint(Sender: TObject);
2154    procedure FormCreate(Sender: TObject);
        procedure Image1Click(Sender: TObject);
2156    procedure Image2Click(Sender: TObject);
        procedure Image3Click(Sender: TObject);
2158   private
        { Private-Deklarationen }
2160   public
        { Public-Deklarationen }
2162   end;

2164 var
      Form6: TForm6;
2166   eichfaktor:real;
     implementation
2168  uses unit1,unit2,unit4,unit5,unit7;
     {$R *.DFM}
2170
     procedure TForm6.FormClose(Sender: TObject; var Action: TCloseAction);
2172 begin
      Form7.close;
2174  Form1.mediaplayer1.display:=form2;
      form2.visible:=true;
```

```
  Form1.Mediaplayer1.displayrect:=Rect(0,0,Form2.clientwidth,Form2.clientheight);
  geeicht:=false;
  eichfaktor:=1;
  eifak:=1;
  Form1.tabelle.rowcount:=2;
  screen.cursor:=crdefault;
end;

procedure TForm6.kalibrierenbtnClick(Sender: TObject);
var ex1,ex2,ey1,ey2:real;
begin
geeicht:=true;
MessageDlg('Bitte clicken Sie auf den Anfangspunkt einer Strecke bekannten Ausmaßes.',
        mtinformation,[mbOK],0);
Screen.cursor:=6;
KalibrierenBtn.Tag:=1;          {1.Eichungswert}
Eix1:=-1;               {Hilfswert für repeat-Bedingung}
Repeat
Application.processmessages;
until Eix1<>-1;
Screen.Cursor:=crdefault;

MessageDlg('Bitte clicken Sie jetzt auf den Endpunkt.',
        mtinformation,[mbOK],0);
Screen.cursor:=6;
KalibrierenBtn.Tag:=2;                                  {Flagge für Unit2 ,2.Eichungswert}
Eix2:=-1;
Repeat
Application.processmessages;
until Eix2<>-1;
Screen.Cursor:=crdefault;
KalibrierenBtn.Tag:=0;

Form4.show;
Form4.bringtofront;
Streckeinm:=0;
Repeat
Application.processmessages;
until Streckeinm<>0;
Form4.visible:=false;

ex1:=eix1;ex2:=eix2;ey1:=eiy1;ey2:=eiy2;
try
   Eichfaktor:=Streckeinm/Sqrt(Sqr(Ex2-Ex1)+Sqr(Ey2-Ey1));
 except
  Messagedlg('Fehler.'+#13+'Bitte wiederholen Sie die Eichung.',mtinformation,[mbok],0);
 end;

Eifak:=Eichfaktor;
if MessageDlg('Möchten Sie noch den Nullpunkt verschieben?',
        mtinformation,[mbyes,mbno],0)=mryes then begin
     Screen.cursor:=6;
     KalibrierenBtn.Tag:=3;                             {Flagge für Unit2 ,Nullpkt}
     Nullpktx:=-1;
     Repeat
     Application.processmessages;
     until Nullpktx<>-1;
     Screen.Cursor:=crdefault;
     KalibrierenBtn.Tag:=0;
     end;
end;

procedure TForm6.DiagrammBtnClick(Sender: TObject);
var i:integer;
    s:string;
    t:boolean;
begin
t:=false;
if Form1.tabelle.rowcount<4 then exit;
for i := 1 to Form1.tabelle.rowcount-1 do begin
  s:=Form1.tabelle.cells[1,i];
  if s='' then t:=true;
end;
If t then messagedlg('Mindestens ein Bild wurde übersprungen! Die angezeigten Diagramme haben damit Fehler',
                mtwarning,[mbok],0);
form5.show;
form5.bringtofront;
```

```
        form5.XYDiagr;
2254    form5.tabellefuellen;

2256    end;

2258    procedure TForm6.SpeedButton2Click(Sender: TObject);
        begin
2260    if Form1.mediaplayer1.position=Form1.scrollbar2.position then exit;
        Form1.mediaplayer1.position:=Form1.mediaplayer1.position-1;
2262    label6.caption:=inttostr(Form1.mediaplayer1.position);
        end;
2264
        procedure TForm6.FormPaint(Sender: TObject);
2266    begin

2268    Label6.caption:=inttostr(Form1.mediaplayer1.position);
        end;
2270
        procedure TForm6.FormCreate(Sender: TObject);
2272    begin
        geeicht:=false;
2274    end;

2276    procedure TForm6.Image1Click(Sender: TObject);
        begin
2278    form7.cursor:=1;
        end;
2280
        procedure TForm6.Image2Click(Sender: TObject);
2282    begin
        form7.cursor:=2;
2284    end;

2286    procedure TForm6.Image3Click(Sender: TObject);
        begin
2288    form7.cursor:=3;
        end;
2290
        end.
```

```
unit Unit7;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms,unit6, Dialogs;

type
  TForm7 = class(TForm)
    procedure FormPaint(Sender: TObject);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormCreate(Sender: TObject);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form7: TForm7;
  eix1,eix2,eiy1,eiy2:integer;
  geeicht:boolean;

implementation
uses unit1,unit2;
{$R *.DFM}
 {$R viacurs.res}
procedure TForm7.FormPaint(Sender: TObject);
var pos:integer;
begin
pos := Form1.Mediaplayer1.position;
with Form1.Mediaplayer1 do begin
play;
position:=pos;
end;
end;

procedure TForm7.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
if Button=mbRight then begin
                    if form1.mediaplayer1.position=form1.scrollbar3.position then exit;
                     form1.mediaplayer1.position:=
                       form1.mediaplayer1.position + 1;
                     form6.label6.caption:=inttostr(Form1.mediaplayer1.position);
                     end
   else begin

    If Form6.KalibrierenBtn.Tag=1 then begin
   Eix1:=x; Eiy1:=y;
   end;
   If Form6.KalibrierenBtn.Tag=2 then begin
   Eix2:=x; Eiy2:=y;
   end;
   If Form6.KalibrierenBtn.Tag=3 then begin
   Nullpktx:=x; Nullpkty:=y;
   end;

   if not geeicht then begin
   if messagedlg('Sie sollten erst den Bildschirm kalibrieren. Einverstanden?',
                    mtconfirmation,[mbyes,mbno],0)=mrno then geeicht:=true;
                    end
     else  form1.zeileinsmemo(x,y);
  end;
end;

procedure TForm7.FormCreate(Sender: TObject);
var i:integer;
begin
For i:=1 to 8 do begin
screen.cursors[i]:=loadcursor(hinstance,makeintresource(i));
end;
end;
end.
```

ANHANG B: Spezielle Anleitung zur automatischen Analyse des Filmes Flummi.avi

1. Starten des Programmes **VIANA.**
2. Laden des Filmes Flummi.avi[40].
3. Wahl der automatischen Analyse, '*Automatisch'*.
4. Kalibrieren des Bilsschirmes, '*Kalibrieren*'.
   Ein Strich auf der Meßlatte (links im Bild) entspricht 5 cm.
5. Einschränken der Filmlänge mit den Laufleisten unter dem Videobild. Soll der ganze Film analysiert werden, so ist dies nicht nötig.
6. Die optimalen Einstellungen sind bei diesem Video :

Toleranz:	40
mögliche Abweichung:	40
Zweittolernz:	100
Objektgröße:	20

7. Mit dem Mauszeiger auf den Ball im ersten Videobild klicken, um eine Suchfarbe zu bestimmen.
8. Starten der Analyse, '*Filmanalyse'*.
   Sollte die Analyse mit einer Fehlermeldung abgebrochen werden, so wiederhole man die Punkte 6 und 7.
9. Öffnen der Diagrammanzeige, '*Auswertung*'

Anmerkung:

Da der Film stark komprimiert wurde, sind die Ergebnisse fehlerbehafteter als bei dem Originalfilm im M-JPEG - Format. Der Flummi ist in dem Video nicht immer rund, siehe auch Abb.74.

[40]Es hat Geschwindigkeitsvorteile, wenn sich Programm und Film auf der Festplatte befinden

ANHANG C: Internetadressen

Über das Internet kann man kostenlos viele interessante, physikalisch relevante Videos und nützliche Software beziehen:

Physik - Filmsammlungen zum Herunterladen:

http://members.aol.com/raacc/data.html *World-in-Motion Data Base*

http://members.aol.com/raacc/wima.html#database ´´

http://152.42.32.64/movies.pub/?table *Davidson Physics Movie Data Base*

http://sunhehi.phy.uic.edu/ ~clive/IP/home.html *Interaktive Physikexperimente*

http://www.physik.tu-berlin.de/institute/IFPL/IBE/IBE.html *Interaktive Physikexperimente Berlin*

Videoanalysesoftwware:

http://didaktik.physik.uni-wuerzburg.de/ ~pkrahmer/home/galileo.html *Galileo Multimediaprojekt*

http://www.Isw.com/videopoint *Programm Videopoint*

Weitere interessante Seiten zum Thema Videoanalyse und Computer:

http://www.cad.uni-sb.de/%7Ealex/software/trmoov.zip *Filmformatkonvertierung*

http://www.public.iastate.edu/%7Estark/gutil_sv.html *Sammlung nützlicher Software*

http://quicktimevr.apple.com/sw/qtvrwin32.html *QuicktimeVR - Treiber*

http://www.glenbrook.k12.il.us/gbssci/phys/mmedia/index.html *Multimedia Physik Studio*

http://members.aol.com/raacc/wim.html *Softwarecodecs*

http://www.miro.de *Verschiedenes zur Digitalisierung von Videos*